Open Source Programming as a Framework for Scientific Collaboration: An Example in the Context of Land-use Change Modeling

Charles M. Schweik

Prepared for the Workshop on Scholarly Communication as a Commons,
Indiana University, Bloomington, March 30-April 2nd, 2004

1. INTRODUCTION

Traditional approaches to the communication and validation of scientific research (e.g., peer-review) and the communication of findings (e.g., refereed publication) have been in place in some form since shortly after the development of the printing press in the sixteenth century (Ziman, 1969; Johns, 2001). This process of peer-review as a mechanism to check for credible information (Burnham, 1990; Kronick, 1990) and journal publication has led to incredible progress in humanity's scientific knowledge over the last four centuries. It also provides an example of how advances in technology (the printing press coupled with systems for the delivery of mail) can change the speed in which scientific knowledge can build.

However the evolution of the traditional process of scientific discourse over the course of four centuries is not without two age-old problems, and one more recent one. One problem relates to the cost of scientific publishing and results in a system that forces brevity in publication, such as a word length limit in the instructions to authors by journals. A second problem relates to the global distribution (or lack thereof) of paper copies of the journal. Interested scientists in some parts of the world (e.g., developing countries) cannot easily acquire state-of-the-art knowledge because of inadequate library (paper-based) resources. A third more recent problem arises as scientific knowledge is based on complicated, technical know-how, often grounded upon computing. For example, in the context of the scientific advances in land-use change modeling (an issue described later in more depth), significant and complex approaches now exist, but there often is slow growth or application beyond the model's original founding group because of the transaction costs (and potentially the expense) to learn and apply the model elsewhere. Moreover, Aber (1997:232) notes that in the field of ecology there is a general distrust in models because these projects and papers are not held to the same standards of full disclosure and peer-review as other areas of scientific research. We would argue that Aber's conclusion is, at least in part, driven by the first problem of brevity in publishing.

An emerging opportunity?

In a paper entitled "Beyond the Information Revolution," Peter Drucker (1999) makes the observation that when new technologies emerge, it often takes years before real innovation utilizing this technology occurs. Drucker uses the example of the railroads in the nineteenth century, where they were originally envisioned only for the transportation of people; according to Drucker, it took thirty years before they were utilized for the transportation of freight.

In the context of scientific collaboration, we may be in such a lag period with regard to the Internet and specifically the World Wide Web. For example, with the exception of one or two emerging innovators,[1] most "e-journals" currently in existence follow the design of the old paper model – volumes and issues, with hyperlinks to papers or abstracts. While this in itself is an important innovation from a developing world perspective (as Internet access becomes more readily available globally), web technologies have the capacity to move traditional methods of scientific communication and collaboration to systems that are much more powerful, immediate, and interactive. Like Drucker's case of the railroads, the argument we make in this paper is that we may have reached a point where we can apply Internet-based technologies and collaborative approaches that have been around for nearly a decade in new ways to enhance the way scientific research is communicated and built upon in a global context. The vision we present here involves the combination of:

(1) The Web as an interactive platform for communicating throughout the research process;

(2) The traditional concept of "peer-review" as a mechanism for maintaining quality control and as a mechanism to encourage participation; and,

(3) "Open Source" programming and licensing and emerging "Open Content" as a model for global, collective action on scientific research.

Most readers will be familiar with the Web (#1) and the traditional peer-review process for evaluating scientific research (#2). Therefore, this paper will concentrate on discussing Open Source (OS) as a potential model for global, Internet-based collaboration and how we might

[1] *Ecology and Society* (formerly*Conservation Ecology*) is one innovative example.

initiate a new form of collaborative scientific research grounded upon the OS software development paradigm.

The paper is structured as follows. First, for readers who have not been exposed to the OS collaborative paradigm, we provide an overview on open source fundamentals. Second, we address the question – Is OS simply a special case of the traditional scientific process? – and conclude that it has similarities but some important differences. Third, we describe our initial thoughts on initiating a collaborative research effort in the context of land-use change modeling, following an OS-like approach. We conclude the paper by raising some important generic issues that will need to be addressed in the future to get such research programs moving forward.

2. A PRIMER ON OPEN SOURCE/CONTENT LICENSING AND PROGRAMMING

Traditionally, software has been developed by individual organizations or firms, is treated as proprietary and is distributed through the sale of software licenses. The buyer is permitted to install the software on one or more computers dependent on the parameters of the software license he or she purchased. The logic of the software – the company's intellectual property – is protected through the conversion of the programming logic (source code) into compiled boolean logic that can be understood and run by a computer but cannot be read by humans. The accompanying software license prohibits the copying or sharing of the software beyond what is permitted by the licensing agreement and the company holds "all rights reserved."

OS software development projects utilize a completely different licensing approach. At the heart of this is issues surrounding intellectual property, and the decisions an author (such as a software developer) makes to license the property that he or she has just created. The choice, traditionally, has been dichotomous: to reserve all rights (copyright) or to make it public domain (freeware). But back in 1984, efforts by Richard Stallman and others who now are part of the Free Software Foundation (http://www.fsf.org/) created an alternative software licensing approach, called the "GNU General Public License." The existence of license, and alternatives to it that now exist (see opensource.org, 2004), demonstrate that the rights as software authors and –

importantly – rights of other “content” developers as well (e.g., scientists, as we will get to below) face a decision that covers a spectrum of rights rather than a simple choice between the two options of “all rights reserved” and “public domain”.

Some readers may be familiar with OS licenses but may not be familiar with the more recent emergence of the more general “Open Content” (OC) licenses. Laurence Lessig and colleagues at CreativeCommons.org have recently created a set of licenses that follow the ideas generated in the software domain but can apply to any intellectual property. They also provide a helpful description of four key questions an author needs to ask and that drive differences in OS and more general OC licenses (CreativeCommons.org, 2004; also see Perens, 1999; Feller and Fitzgerald, 2002 for related discussions). These questions are as follows:

- *Is it OK for others to freely copy and distribute this intellectual property?* If yes, users can download and freely distribute the software – sometimes referred to as “copyleft”.

- *Can derivative works be made based on the readable source code?* In open source licenses, the readable program source code must be made available and the license then stipulates whether others can make alterations to it.

- *If derivative works are permitted, what are the rules for distribution of these derivative works?* Many OS licenses require that any derivative work be given back to the community, meaning that it falls under the same licensing scheme. This is sometimes called a “viral” license (Pavlicek, 2000), because its rules get passed on to other derivations of an original work. If an author answers no to this question, the license may allow the derivate work to be included in a commercial software package.

- *Is author attribution required?* Answering yes to this question requires the user of software or content to give credit to the original author(s) who developed the work. In software development settings, as the software evolves, this requires some mechanism to document intellectual property contributions as new derivative works (versions) evolve. In programming settings, this is often done in the “header” of programs where contribution histories are listed sequentially.

Answering each of these core questions (yes/no) create a situation where authors (software developers, writers, musicians, or scientists) can decide which rights they wish to preserve and which rights they wish to wave related to their intellectual property (what Lessig calls “some rights reserved”; Stix, 2003). The endpoints on this spectrum are “all rights reserved” or copyright, and “public domain” at the other end. In the broader domain of Open Content, this has led to a suite of 11 different licensing variants based on how these questions are answered (Creativecommons.org, 2004).

This suite of licensing options provides at least two *potential* advantages over the traditional copyright approach to software development. First, OS software is provided at no cost to the end user, thus providing a major incentive for it to be utilized, especially by people and organizations that are working under limited budgets. Nonprofit organizations, some government entities and individuals (e.g., college students) are important communities who are embracing OS initiatives (see Hahn, 2002). Second, depending on the application, there is the potential to generate a large community of users and developers globally – larger than any one proprietary organization could create – to work on future versions and enhancements of the software.

Thousands of OS software are currently under development. At perhaps the premier OS development website, Sourceforge.net, there are over 70,000 hosted OS projects and nearly 800,000 registered users (Sourceforge.net, 2004). It is likely that the majority of these projects are dead or dormant and probably the work of students who have uploaded some college project. But even if 10% of these are active, this is substantial. The Linux operating system and the Apache web server software are perhaps two of the oldest and most prominent examples of OS success. They can be considered "enterprise" software in that they have achieved a significant global "market share" and amassed a substantial (thousands if not hundreds of thousands) in their user and development community. We expect that a major reason for the emergence of these enterprise OS programming projects is because they are embraced not only by volunteers but also by science and technology companies and government agencies. For example companies such as IBM (Linux, IBM-Eclipse software project), SUN Microsystems (the NetBeans project and OpenOffice.org project), Hewlett-Packard (HP-Gelato project) have full-salaried employees contributing to open source software development projects (Gasser and Scacchi, 2003). Similar activities are occurring in some government agencies as well. For example, some of the software used to visualize data acquired by the NASA Mars rovers Spirit and Opportunity rely on open source components (Norris, 2004). Brazil's National Institute for Space Research provides another example, where they have devoted significant resources toward the development of open source geographic information systems software called “TerraLib” (NIPE, 2004).

3. IS OPEN SOURCE SIMPLY A SPECIAL CASE OF THE TRADITIONAL SCIENTIFIC PROCESS?

Some have suggested that OS collaboration is a special case of the traditional approach to scientific discovery (Bezroukov, 1999a; 1999b). The case could be made that the traditional method of submitting papers to refereed journals, and publishing one’s findings are similar to the OS process. But in our view, there are some important differences (Schweik and Semenov, 2003).

First, in OS collaboration the entire research product is shared with the community, including the research process, rather than just a summary of the end results. In traditional publishing, space limitations in journals limit what can be provided to the community. Second, OS collaboration over the Internet increases the size of the community who might utilize the material. Internet access continues to expand globally. Lawrence (2001) reports that articles freely available on the Internet are cited more widely. This suggests that information freely served on the Internet can reach larger audiences if it is well marketed. Third, the OS licensing of free sharing and distribution of intellectual property (e.g., copying is allowed – even encouraged, but with author integrity intact) is a different model of intellectual property rights than the traditional copyright system used in most scientific publishing. Fourth, OS collaboration over the Internet potentially increases the speed in which innovations can be published. “Publishing” (e.g., improvements to a program module) are made incrementally, and in some situations the process of peer-review could be nearly immediate.

Our main argument is this: *The convergence of three phenomena – (1) the ease in which geographically distributed researchers can now collaborate over the Internet because of such technologies as email and the Web, (2) the development of infrastructure to support global collaborative infrastructure for Open Source programming, and (3) Open Source/Content licensing – has tremendous potential to create dramatic increases in* *humanity's ability to solve complex problems in almost any scientific research or collaborative endeavor*.

Lessig's (2001) observations on the Internet as an "Innovation Commons" provide evidence to support this statement. In his book, *The Future of Ideas*, Lessig argues that two components of Internet design led to the exceptional creativity and innovation we have witnessed over the last decade. The first is the fundamental network architecture principle called "end-to-end" design. The idea behind that kind of network design is to keep the "intelligence" in the network at the "end points" (the software on the client computers or Internet servers) and to leave the network transmission infrastructure relatively simple (Lessig, 2001: 34). This simplicity in transmission allows for tremendous innovations to occur at the end-points (e.g., the client computers that people use to gain access to the Internet). In the view of the U.S. National Research Council, "end-to-end" design was a key component to the development of Internet services and software because the data transmission over the Internet was simple, standard and nondiscriminatory (Lessig, 2001: 40).

Lessig notes that the second important design component of the Internet, or specifically, in this case, the Web, was the decision to include the "view source" option in web browser menus. This feature made it very easy for anyone with Web access *to learn* from others on how to develop web pages (Lessig, 2001: 57) by reading and copying the source. Especially in the days before HTML writing software like Macromedia Dreamweaver™ or Microsoft Frontpage™ came along, the Web grew at the incredible pace it did largely because of this capability; others could view, learn from, copy and implement web source code written by others through this "view source" browser function.

This brings us to a critical point that is typically not recognized: while not licensed as such, the growth of the Web between 1996-2000 *was* an OS phenomenon at a global scale, and given the rapid growth of the Web during this period it could be argued that it was (and is), by far, the most successful Internet-based Distance Learning program in history. The OS aspect of browsers and the Web led to a tremendous amount of innovation – perhaps the fastest and broadest innovation in the history of humankind – and it provides a critically important example of the kind of innovation that can be generated through the Internet combined with OS approaches.

In short, it is our belief that the convergence of the end-to-end design, open source/content licensing, and principles of OS project design (discussed in other work we have done, see Schweik and Semenov, 2003), lead us to new opportunities in the way we may undertake scientific research that have real potential to increase the rate in which new discoveries or advances are made. Drucker's (1999) discussions on the lag of innovation help to keep us aware not to get stuck in old ways of thinking. We now turn to a description of how we might build an open source-based collaboration in the context of one scientific research domain: Land-use change modeling. The description that follows focuses on this particular area of scientific collaboration, but many of the issues we are contemplating are in many instances generic to other scientific areas as well.

4. APPLYING AN OPEN SOURCE COLLABORATION APPROACH IN SCIENTIFIC RESEARCH: THE EXAMPLE OF LAND-USE CHANGE MODELING

Land-use change modeling is an area ripe for such an OS/OC collaborative endeavor for several reasons. First, there is substantial (global) interest for these tools. For example, scientists and government officials need such models to better understand how the land-use system functions and to explore the effects of various policy or planning initiatives (Verberg, et al. forthcoming). But model development is hampered in part because of difficulties researchers have in building on others' work. Recent reviews of available models sponsored by the U.S. EPA (U.S. EPA, 2000) and by the U.S.D.A Forest Service (Grove et al., 2002; Agarwal, et al., 2002) have identified more than 30 different models currently available (some in the public domain, some commercial packages), and each using different approaches and technologies (including many that are Geographic Information System-based). While having a relatively large number of land-use change models is beneficial in terms of innovation, building upon this prior work is difficult for several reasons: (1) these models often require special disciplinary expertise that may not be available in a particular organization; (2) these models are often technologically and theoretically complex, require extensive data; and (3) only limited information on models is typically available through traditional publishing methods. Consequently, should someone want to try and learn and apply the model to another geographic location, the transaction costs to do so can be substantial. And based upon our recent reviews of the literature (Grove et al, 2002; Agarwal et al., 2002) we get a sense that there is significant room for improvement by the scientific community to build on each other's work. Most of what we see on the evolution of individual models over time come from the same groups that initially developed the models.

Parker and colleagues (2002: 212) touched on these issues and others in their discussions on the challenges of integrated assessment and modeling. They note that some of the barriers to advancing models (such as land-use change models) are the separation between scientists in difference disciplines, separation between scientists and modelers, separation between application modelers and software interface designers, separation between scientists and decision-makers and the community, and fragmentation in education. For all of these reasons, we believe that building a land-use modeling effort based around an OS collaborative paradigm might lead to some real productivity enhancements in this area of scientific research and provide a mechanism to close some of these gaps between stakeholders.

But just how might it be accomplished? Building on our previous work studying how open source programming projects operate (Schweik and Semenov, 2002) and our ongoing efforts to initiate such an "open source/content" collaboration (see Schweik, Evans and Grove, 2003), what follows is a discussion on some important issues that need to be considered in order to establish such a collaboration.

4.1 Two Example Models: UrbanSim and CLUE

To better explain our vision of how this collaborative paradigm might be applied to people unfamiliar with the area of land-use change modeling, let us first present a brief overview of two prominent land-use change models, one operating at the spatial scale of a metropolitan region (UrbanSim) and the other operating at a broader spatial scale of one or more national boundaries (CLUE). For brevity, we will only provide an overview of the general approach and structure of the models to show how an open source/content paradigm might be applied to further develop each. Readers interested in more detail on UrbanSim are encouraged to read Waddell (1998, 2000), Waddell and Ulfarsson (forthcoming), and Noth et al. 2003. More information on the CLUE suite of models can be found in Veldkamp and Fresco (1996a, 1996b, 1997), de Koenig et al. (1999), Verberg and Veldkamp (2001), and Verberg et al. (1999a, 1999b, 2002). This summary is based upon reading these articles. Both models are spatially explicit and temporally dynamic, and involve components that require quantitative methods.

UrbanSim

According to their project website (www.urbansim.org), "UrbanSim is a software-based simulation model for integrated planning and analysis of urban development, incorporating the interactions between land use, transportation, and public policy" and is designed specifically for use and application in metropolitan regions. Consequently, UrbanSim models land-use change at fine spatial resolutions, usually with one cell in their spatial grid representing a 150 by 150 meter area. Recent empirical applications of UrbanSim include Eugene-Springfield, Oregon, Salt Lake City, Utah, and Honolulu, Hawaii. UrbanSim explicitly models the location choices made by households (housing), businesses (location and jobs), and the location choices of real estate developers. It utilizes a set of interacting sub-models representing these different actor types and corresponding processes in urban environments. Discrete choice modeling (e.g., multinomial logit) is utilized to predict location choices. Geographic Information Systems is used to integrate data and display model results.

Figure 1 provides a graphic of the general structure and processing of UrbanSim (Waddell, 2002). Inputs to the UrbanSim model include the initial year data (e.g., current land-use configuration), data from regional economic forecasts provided by an externally developed macroeconomic model, travel access indicators provided by another externally developed travel demand model, and user specified data that provides input for various public policy scenario investigations. This information gets fed into a "model coordinator" computer program that coordinates the processing and dialog between five sub-modules: Accessibility, Economic and demographic transition, Household and employment mobility, household and employment location, and Real estate development. The Accessibility model predicts the pattern of accessibility by auto ownership level. The Economic and Demographic transition estimates the creation or loss of households and jobs by type. The Household and Employment Mobility sub-module organizes movement of households or jobs within the region. The Household and Employment Location sub-module models the location choices of households and jobs from the available vacant real estate. The Real Estate Development sub-module determines the location, type and quantity of new construction and redevelopment by developers. Applications of UrbanSim tend to model changes in residential housing and business (employment) locations over a 15 to 20 year period at an annual time step. The UrbanSim computer application is one of the only land-use change models we have identified that is already licensed as OS.

*** Figure 1 about here ***

CLUE

The general CLUE approach to modeling has been applied to study land-use change in many “developing world” contexts, including Ecuador, Costa Rica, China, Indonesia, and the entire Central America region. A primary focus is to understand change in agriculture land-use in these countries. It follows a dynamic, spatially explicit, quantitative and multi-scale approach to land-use change modeling, and assumes that land-use is determined by a combination of biophysical and socio-economic driving forces. CLUE models generally are run for approximately a 20-year duration at annual time steps. The CLUE modelers distinguish between driving factors that determine the quantity of change annually versus drivers that determine the location of change (Verberg et al. 2002). Consequently, the modeling approach is comprised of two primary subcomponents: (1) the Demand Module and (2) the Allocation Module (Figure 2). The Demand Module determines for the spatial extent of interest, the demand for a given year for all of the various land-use types. The allocation module takes as input the area of various land-use types required for that particular year from the Demand Module, and uses derived multiple regression equations from past land-use change, to geographically allocate new land-use demands.

These two land-use change modeling approaches demonstrate the multi-disciplinary nature and level of sophistication (technically, theoretically, empirically) involved in current-day land-use change modeling. Both require data gathering, formatting, and a variety of steps (some done manually by the modeling analyst, some through automation such as the model coordinator in UrbanSim) that need to be taken. These approaches are quite similar to OS programming efforts, in that in order to make some improvements or alternations to a sub-module, some programming skills may be required (at least in the case of UrbanSim, it is unclear from our readings on CLUE if that is also the case). But from the standpoint of making scientific advances in the area of land-use change modeling, moving to an OS approach extends the paradigm to new areas. Intellectual contributions are required not only by modeler/programmers, but also by theoreticians and scientists from a variety of disciplines (economics, geography, political science, sociology to name a few) interested in the processes of land-use change, as well as practitioners such as policy analysts and planners who are interested in what information or scenario projections these models may reveal for a geographic area of interest. Moreover, as literature from both of these models suggest, as these models get applied in various regions of the Earth, different driving variables of change and processes will become important. This suggests that if applications of these models in other places expand, the logic within various sub-modules will need to be modified to capture these process differences. For example, we would expect that some distinct differences in land-use change would exist if studying an area of South America compared to South Asia at either broad (e.g., national) or local (e.g., metropolitan) scales.

*** Figure 2 about here ***

4.2 The General Vision of Open Source/Content Land-use Modeling

Building from the OS programming paradigm, our general idea is this: Imagine that the model developers of UrbanSim or CLUE (or one of the many other land-use change models currently in existence) decided to license their models under an open source licensing scheme as described earlier (e.g., copying is allowed, derivative works are allowed, new derivative works fall under the same open source license, and author attribution required). Imagine that these developers make all the source code to the models, the documentation, their empirical data, and any subsequent papers all available over the Internet and all of these products fall under open content licenses. Important additional documentation might include “how to” manuals or distance learning material explaining to new or relatively new users or interested future modelers how to run and add new contributions to the model. In the case of UrbanSim, this would mean providing documentation on each of the sub-modules in Figure 1 and, in a pure open source model, even the model coordinator. The same would be true in regard to the processes and procedures to undertaking a CLUE modeling approach and with its Demand and Allocation sub-modules.

By making all of this licensed as open content and available over the Internet, someone interested in urban modeling in Russia (for example) could conceivably download and borrow all the UrbanSim logic, and make appropriate modifications to its structure that makes theoretical sense for a Russian context. For example, there could be substantial differences in the logic of the Economic and Demographic transition modules for an application in the United States versus a Russian context. But the major point is that through this alternative licensing approach, the ability for someone to find, borrow, learn and apply the model to another region of the world and contribute new logic may be substantially increased, compared to the methods in which knowledge and modeling approaches are currently shared (largely through traditional publication mechanisms where there are substantial limits in what can be published). But this vision is much easier said than achieved. We will now turn to a discussion of some critical issues that would need to be addressed in order to make such a vision viable.

4.3 Attention to Incentives for Participation (Issue 1)

One of the real puzzles about OS programming projects has been the phenomenon of large groups of global volunteers donating their time and intellectual property to the project (Glass, 1999). The question of why do programmers essentially donate their time and intellectual property is perhaps the most studied and the best understood aspect of OS projects. The prospect of OS participation has puzzled economists in particular (Lee, Moisa and Weiss, 2003; Ghosh, 2003) because, at least at first glance, it goes against theories that rational self-interest motivates individuals (when people forgo direct compensation for providing work and intellectual property to the project), and it challenges theories of economic organization (e.g., firms and markets). Studies have identified reasons for participating, some of which are motivated by longer-term economic desires (Lakhani, et al 2002; Feller and Fitzgerald, 2002; Hertel, Niedner and Herrman, 2003). These motivations include: meeting some personal software need; addressing a software crisis; pure intellectual stimulation; the desire to belong in some community; contributing to some specific social movement (software should be free!); or simple altruism.

But recently, literature from labor economics theory – human capital theory and signaling theory – helps to explain OS voluntary participation (Hann et al. 2002). From a human capital standpoint, participation in open source programming projects in a “virtual” and in some cases nearly “global” team, allows actors to *learn* and gain valuable skills through the process of reading others’ software code and receiving feedback from their own contributions through peer-review mechanisms. Peer-review in OS projects is vital for quality control, learning and recognition of ones work (Voightmann and Coleman, 2003). In this regard, contributions are seen as future-oriented investments in building one’s career (Lerner and Tirole, 2002a; Johnson, 2002; Lee, Moisa and Weiss, 2003). Moreover, high quality contributions can act as a signaling device to the rest of the community of a participant's programming abilities (Lee, Mosia and Weiss, 2003) establishing a reputation (Sharma, Sugmaran and Rajgopalan, 2002), which could lead to work opportunities in the future (Lee, Moisa and Weiss, 2003).

Up until now, the primary focus on OS participant motivations has been individual volunteers. But Ghosh and colleagues (2002) report that nearly one-third of the OS software developers surveyed are directly paid by employers to participate in the project. Wichmann (2002) reports that most of the 25 firms they identified as active in OS projects to participate in Linux development were largely motivated by self-interest – e.g., product standardization, cost savings, strategies to weaken competition, and efforts to make their own products compatible with OS products.

What can we take from all this motivation research on open source programming if we are to borrow the collaborative paradigm and apply it to a broader scientific collaboration, such as land-use change modeling? First, a comparison between the motivations between OS programmers and scientific researchers are not all that different (Schweik and Semenov, 2002). There will be many in academia (graduate students, junior faculty, even more senior faculty) who would want to participate in a broader “virtual” endeavor in order to build or hone new skills. The distance-learning component of an OS-like collaboration where one reads other peoples “source” (models, papers, etc.) and having submissions of their own go through peer-review and the learning that comes from that process can be a big motivation for many.

Second, the motivation to “signal” one’s abilities and skills (e.g., getting one’s name known) to others active in the same scientific domain is another motivation for scientists, particularly for new (e.g., junior faculty at universities) or emerging scientists (e.g., graduate students). In a recent workshop we held on this OS land-use change idea (Schweik, 2004), one land-use modeler said (paraphrasing): “I would have gladly made my model open source in graduate school. That way others might have used it and it would have gotten my name more widely known.”

But “signaling” leads to an important concern. In order to signal one’s abilities by posting intellectual property, one’s name needs to be associated, over time, with that submission. Key to motivating people to submit in open source/content collaboration will be *serious attention to author contribution tracking*. In the infrastructure built to support this collaboration (discussed more below) there needs to be real attention to how someone contributed over time to a new model module, to land-use change theory, to empirical findings, or to other project content.

Finally, in order to encourage participation in the scientific domain means that we have to pay close attention to what drives scientists to contribute and collaborate in the current systems and job environments they work in. For many scientists, having their work published in high quality, refereed journals is a key measure of “success” in their field and a measure often used for job promotion. This is particularly true in situations where a junior academic is working toward tenure and could provide a significant disincentive to contribute to an open source scientific collaboration. Consequently, in the long term, any OS/OC scientific endeavor must consider developing what we call the “next generation e-journal” that would require any component of the product to be “published” only after some level of peer-review and acceptance.

The reason we call this a “next generation e-journal” is because it would not only publish traditional content (e.g., papers on theory related to land-use change or the results of empirical studies applying a particular model), but it would also publish new or revised versions of complete models or sub-component modules (e.g., Figure 1 and 2), and also continued development of distance learning material related to using or contributing to a model. In some instances, even datasets that models use might be “published” after some level of peer-review (such as a dataset on economic projections for a country that might be utilized in another application of the model). In other words, all components of model development or application could be published in this e-journal, which is vastly different than the traditional 30 or so page “final results” paper that is published in the standard paper journals (and many or most current e-journal systems).

This idea is a radical proposal and one that is difficult to change in the current scientific culture. It means a change in what we consider what is “publishable.” But in our view, lots of intellectual effort goes into writing the logic in a land-use change model, and consequently given the ability to post these things on the Web, all research products could be considered potentially publishable! By having this kind of “source” peer-reviewed and available under an open source license on the Web, it has the potential of greatly enhancing the scientific community’s ability to build on one another’s work.

4.4 Project Infrastructure (Issue 2)

A second important issue to moving this idea forward is the development of project infrastructure to support group collaboration. This links directly to the longer-term idea proposed above related to a next generation e-journal, for this infrastructure we discuss now is, in fact, the critical components or functions that would be parts of this next generation e-journal.

In current-day OS programming projects, group collaboration is supported and controlled through Web-based communication and Version Control Systems. For example, the website www.sourceforge.net (discussed earlier) freely hosts OS programming projects and provides group communication functions and software version control systems based on another open source software called “Concurrent Versioning System” or CVS.

Several functions CVS provides will be important to support an open source land-use modeling effort, or any scientific collaboration based upon an open source/content licensing approach. The primary functions in CVS (and other version control functions) are to:

1. Archive versions of software
2. Allow for the retrieval of modules
3. Protect against the problem of people overwriting each others new enhancements
4. Document changes over time (author tracking)
5. Provide analysis functions to identify differences between module versions

A similar set of functions would be required to support the land-use change modeling effort, but would need to be able to support the various broader set of submission types such a project would bring. For example, consider the previous example of the UrbanSim model in Figure 1. Researcher A might be interested in applying UrbanSim for a city that has more land at the fringe of the urban/agriculture interface. He or she may first decide to write a more theoretical paper on the factors leading to developer choice for locating new residential development for this empirical context. (It may be that there are differences in these factors for the mid-west United States compared to the West coast U.S., for example). He might then wish to submit for review this new paper as part of the “theoretical” or “paper” component of the project, thereby using a function similar to the posting of enhancements of software (number 3 above) but for papers or “documentation” (something that existing versioning systems like CVS can already handle). The author could decide to license this paper as “open content” under one of the 11 Creative Commons licenses (Creativecommons.org). The paper could then undergo a peer-review process and then eventually be rejected or accepted and added to the “production” project library or e-journal.

Similarly, this (or another) researcher with sufficient skills may then decide to take the logic described in the theoretical paper and implement a new version of the “household location module” of UrbanSim (Figure 1). This would require him or her to be able to retrieve relevant modules and then, assuming these modules fall under an open source license where derivative works are OK, upon completion submit the new version of the module with new logic back to the project for peer-review. Upon a positive review under some project governance structure (discussed more below), this new module could then be added to the project to replace the older module or as a parallel "sibling" module. Future users of the UrbanSim module might then select between the "west coast" location module or the "mid-west" location module. Again, critical to all of this is some mechanism to keep track of intellectual property contributions (author integrity) for all types of contributions – papers (theoretical, empirical), modules, and other project documentation, including distance-learning materials.

In the context of many programming projects, author integrity is maintained through a historical "log" in the main header (comment area) of the computer program. Author contributions are simply tracked through this text log. But following the vision of the next generation e-journal we are describing, better, more sophisticated methods of author contribution tracking will be needed. This is an area that systems like CVS or OS programming projects may not have needed to consider but is vital in order to encourage scientific participation. These author-tracking systems will need to develop a way for authors to "cite" their contributions, and given the speed in which innovations could be published, it will probably require a new method of citing similar to how people cite web pages currently. That is, publishing will move beyond a "journal, year, issue, number" format and require additionally a "date and time" component.

In addition, there will be an opportunity to develop new measures of the significance of someone's contribution. For example, consider two separate sub-module submissions (Module A and B) by two different authors for UrbanSim. Both modules do different functions for the land-use change model they are embedded within. Given the OS nature of the submission, over time, Module A gains wide interest by other modelers or stakeholders using UrbanSim, and several new derivatives of this sub-module are submitted by others as well as several new more theoretical papers related to its logic. Module B on the other hand, spurs little new interest and no new derivatives. It could be argued that the original author of Module A provided an important intellectual contribution and the use or derivative works should cite that original contribution. There may be ways to use hypertext technology or other version tracking functions to better record these kinds of intellectual contributions in a next generation e-journal that could then provide hard numbers that people could place in their CVs.

Finally, we expect the distance-learning infrastructure for the project to be critical and therefore careful attention needs to be made for this project component. We have already discussed the importance of learning as an important motivation for many to participate in open source projects. We envision the design of a next generation e-journal to support the land-use modeling collaboration will need to have distance learning prominent in its homepage web presence. Material would include general documentation on the modeling approach, more detailed literature on the steps to running the model, details on model subcomponents (e.g., modules), frequently asked questions, etc. All of this material could fall under an open content license (Creativecommons.org, 2004) such that if participants or users of the model make enhancements to this documentation, it would be required to be submitted back to the project for peer-review and possible eventual publishing on the website. The project website or e-journal might include, as another incentive for participation, a prominently displayed list of "key enhancements needed" that might serve as a place where graduate students could find potential thesis topics.

4.5 Project institutional design and governance (Issue 3)

A number of recent studies, including several of our own, have begun to focus on project governance as an important component of OS programming project success. However, currently there is very little published on the governance of open source programming projects, and much of what has been is based upon a short set of high-profile OS success stories. For instance, several studies on Linux, describe the lead developer as a "benevolent dictator," working with a team of "trusted lieutenants" who are recognized experts in a particular domain (Shaikh and Cornford, 2003; Moody, 2001; Dafermos, 2001; Sharma, Sugumaran and Rajgopalan, 2002). A quantitative study of OS projects by Ghosh, Robles and Glott (2002) reports that the majority of OS software projects are directed by a single "lead developer" who maintains a centralized decision making structure. Alternatively, Jorgensen's (2001) study of the FreeBSD OS project identifies a nine-member team heading the project, who are elected by and from all developers with authority to post changes. But as Bezroukov (1999b:17) notes: "…in each [OS] project in particular, there are political systems with corresponding and sometimes fuzzy hierarchical structures." Some early studies suggest that decisions related to project direction are reached by consensus (Fielding, 1999; Markus et al., 2000; Mockus et al. 2000). Established systems of rules, shared norms of behavior, voting systems and monitoring and sanctioning systems appear to be important in some OS projects (Sharma, Sugumaran and Rajgopalan, 2002; Schweik and Grove, 2000; Schweik and Semenov, 2003). Each OS project includes conventions that all must follow and some of these enhance the effectiveness of Internet-based coordination (Bonaccorsi and Rossi, 2003).

Consequently, an important issue in extending the OS paradigm to scientific research collaboration will be designing the decision-making structure and system of operational rules for the virtual team. In our workshop trying to initiate an open source land-use change modeling effort (Schweik, Evans and Grove, 2003; Schweik, 2004) it was proposed that the initial set of collaborators would be subdivided into subcommittees: the "modeling" sub-group (e.g., UrbanSim, etc.), the "data" sub-group (responsible for developing standards for metadata, for example), the "theory" sub-group, the "distance learning" sub-group, and a "project administration/governance" sub-group. Some system of rules will probably need to be developed for peer-review of products developed in each area, and perhaps some mechanism to guide conflict management. A good example of major conflict in OS programming projects is a phenomenon called "forking," where the virtual team splits into rival camps because of a major conflict over project direction or strategy (Feller and Fitzgerald, 2002). Because of the open source license (derivative works OK) the team can splinter into two groups who now are working on two distinctly separate projects with different agendas. There will need to be some consideration on how to handle this kind of situation in a scientific research project. But in short, it is conceivable that the governance structure may ultimately become a combination of how a journal is run and organized (editor and editorial board) and how open source programming projects are organized.

A related “institutional design” issue is consideration of the type of license applied for various project output. For example, we are emphasizing that some of the project output (e.g., land-use change models and sub-modules) should be placed under some form of open source license. One of the first priorities of the virtual team above will be to define, for each product area, the appropriate license for project output. For some land-use change models it might be decided that all related products (e.g., the model modules, their documentation, data, and even theoretical papers) could fall under some open source or open content license (creativecommons.org, 2004), permitting new derivative works to be built from them.

For the area of theoretical papers this provides a particularly interesting problem. It is not inconceivable that an economist, for example, writes some theoretical paper about the decision-making process of an Indiana farmer. This logic might then serve as the foundation for logic in some component of an agent-based model of farmer decision-making. Imagine then, a Brazilian researcher wishing to apply the same model to the Brazilian context. If the theoretical paper was licensed as open content and allowing new derivative works, a Brazilian scientist could download the paper, and revise various sections to reflect what he or she believes to be more accurately depicting the logic of a Brazilian farmer’s decision-making process. This could then be input to a new version of the agent-based farmer model, for a Brazilian context. The result would be two separate theoretical papers, one built from an earlier version, and similarly two different model sub-modules of farmer decision-making. Version control and author tracking systems would need to be devised to ensure that the original and subsequent author’s intellectual contributions are tracked over time. This is crucial because this treads dangerously close to the issue of plagiarism. In part because of this, in other research situations, it might be decided that it is allowable to derive new works from earlier models, but people should not be allowable to derive new works based on older content of theoretical papers. In other words, the model might have an “open source derivative works OK” license, but the project papers might fall under more standard copyright licensing. The set of licenses for all research projects is something that needs to be decided by the group initiating the project.

4.5 Project finance (Issue 4)

The last issue we will raise relates to the financing of such OS-based scientific research projects. Much of the early literature on open source programming projects emphasized that largely a body of volunteer programmers initiated and contributed to these projects. Many of the 70,000 open source programming projects residing on sourceforge.net fall under this category and are “financed” through the in-kind donation of programmers’ time on off-hours from their day jobs. However, a recent study by Ghosh et al. (2002) report that roughly a third of their surveyed open source software developers are being directly paid by their employer. Wichmann (2002) describes some of the motivations for firm’s to participate in such activities, which include: (1) the quest for software standardization and to participate in the shaping of the software toward its particular interests; (2) the need to save money in software purchases; and (3) strategic considerations, such as investment in open source software that somehow complement their other products or compete with rival software companies. And the list of companies that participate in open source development is not small. Of the top 25 companies in Software Magazine’s 2001 Software 500 list, Weichmann (2002) reports seven as “major” contributors to OS projects, including IBM, Hewlett-Packard, Compaq, Hitachi, and Sun Microsystems.

We have been unable to find any literature addressing the question on the importance of direct firm or organizational support for the long-term viability of open source software products. The question is: Can open source software projects remain viable over the long term relying solely on volunteers’ efforts? Or does there need to be some degree of financial support paying for, at a minimum, the project administration? Gilberto Camara, of the Braizilian National Institute for Space Research, raised this question in a presentation at UNESCO, when he argued in his presentation (paraphrasing) “the only way open source software projects will be successful is if a government supports the endeavor” (Camara, 2003). While it is an empirical question yet to be answered, it is likely that (at least for major projects) some level of financing will be required. For example, the argument could be made that the long-term success of some of these enterprise OS software projects (even Linux perhaps?) is because of the investment some of these companies are making into the projects.

The same question can be raised when one extends the OS paradigm into the area of scientific collaboration. However, in the context of academic contributions, it seems that some of the “industry support” may already be established. To some degree Universities do pay their faculty to undertake research that contributes to a larger research program. If a faculty wanted to participate in a virtual OS collaboration, most universities would be fine with this as long as the faculty was meeting traditional measures of scholarship and productivity (such as publications in refereed outlets). (However, this may be changing as some Universities look to capitalize on the intellectual property of faculty through venture capital initiatives.) But once again, assuming a university is supportive of standard publishing, this brings us back to the importance of designing an open source/content research program that supports a next generation peer-reviewed e-journal.

This leads to the question of how that e-journal might be financed. We do not have good numbers on the costs of traditional paper journals or even for current e-journal publications. But it is likely that this is a sizable cost for administration (e.g., editorial support, management of the peer-review process) and final printing costs, perhaps in the hundreds of thousands of dollars. Moving everything to a web-based platform may save some costs (e.g., printing) but the peer-review structure (models and model documentation, theoretical and empirical papers, data, distance learning documents) will raise the level of time required by reviewers and various component editors and raise the costs of these activities. Moreover, the infrastructure to support such a journal (described above) is probably more sophisticated than standard e-journal infrastructure in place today. It is an open question whether traditional journal publishers would embrace such an idea. However, an interesting question for further research would be to review the various business models in open source software (e.g., Weichmann, 2002) and to see whether any might be transferable to the process of publishing scientific journals.

5. CONCLUSION

The initial argument set forth in the introduction of this paper was that the convergence of the Web as an interactive platform, the traditional concept of peer-review, and the innovations in “open source” (and now, open content) licensing creates a potentially new paradigm for the production of scientific research, with potentially, contributions provided from scientists across the globe. Up until now, there has been much interest in the phenomenon of open source programming, but few have come to the realization that the collaboration paradigm could be applied to *any* intellectual endeavor – not just computer programming – involving multiple people. In our view, this could be the most important contribution that the open source licensing innovation may produce.

What we have tried to do in this paper is describe in more detail what this means and how an open content collaboration in scientific research might be accomplished. To do this, we have described in some detail our ideas on how this could be applied to one area of scientific research: land-use change modeling. This description raised some generic issues that will need to be addressed as scientific and scholarly communication moves beyond the traditional form of communication (e.g., journals with volumes and issues) to a more complete repository of scientific production where the full suite of intermediate and final research products are all treated as a kind of publication.

We expect the next decade will produce a variety of experiments in the domain of “open content” publishing over the web (indeed, we are seeing some of these experiments already, such as the MIT Open Courseware effort; and see Schweik and Semenov, 2003 for a list of some others). Some of these will work, some will not. But we think of critical concern as these “experiments” move forward will be attention to several issues we’ve described above:

- Incentives for participation. How the open content idea links to the traditional publishing ideas of “peer-review” and journals;
- The design of next generation e-journals, with particular attention on how to maintain and record author integrity;
- Project governance. How do the standard editorial structure found in journals combine with project governance found in open source programming projects? And,
- Finance. To what degree do such efforts need long term financing? Who might finance such efforts? How might open source business models in industry (such as IBM) inform the publishers of journals?

These are questions that might be worthy of discussion in this Workshop.

Figure 1: Structure of the UrbanSim model

(adapted and interpreted from Figures 3 and 4 in Waddell, 2002)

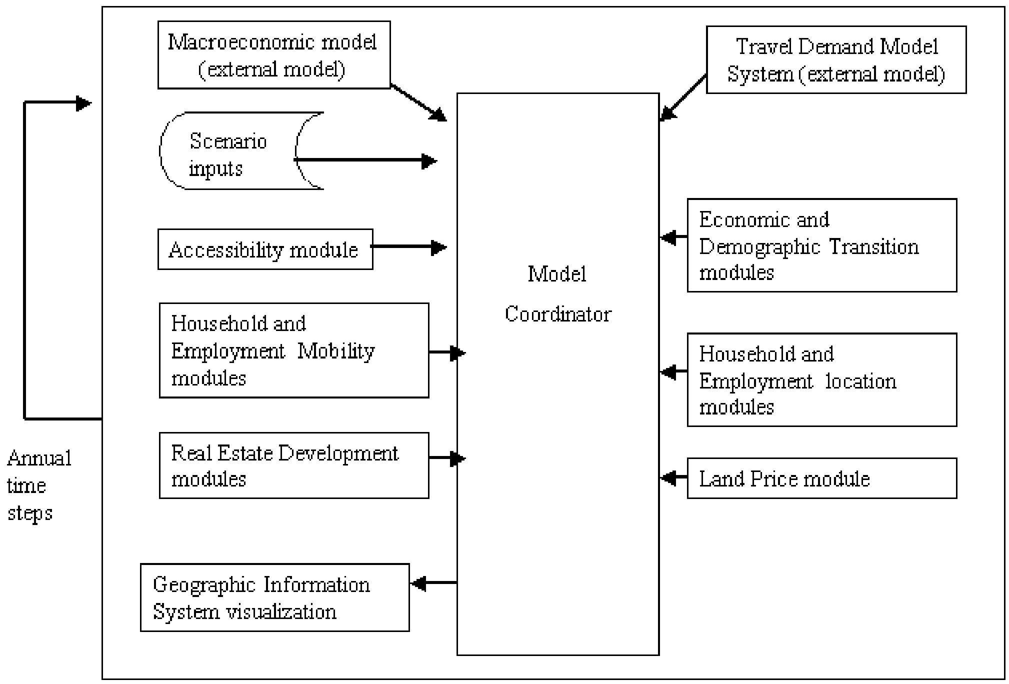
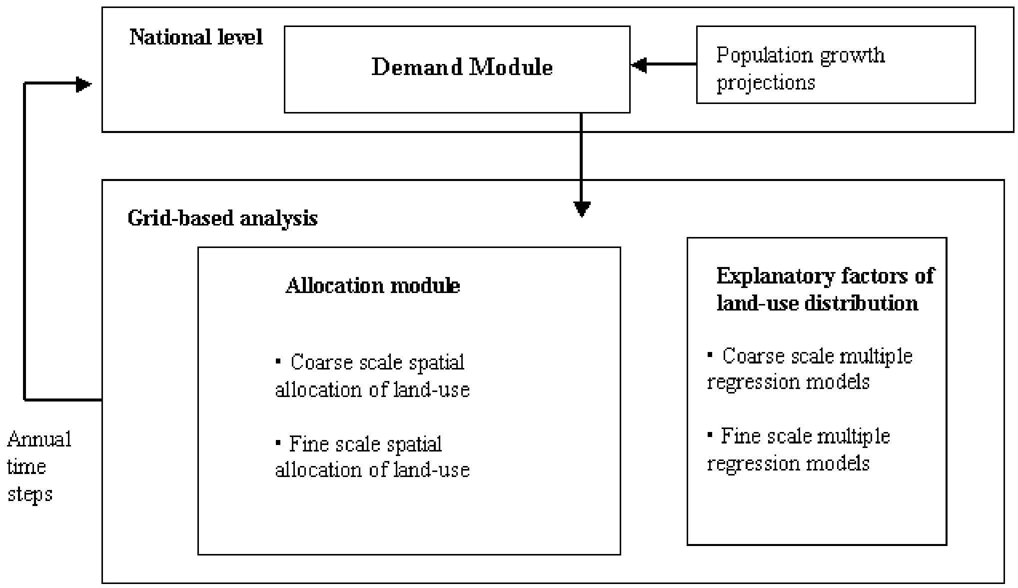


Figure 2: Structure of the CLUE modelling framework
(adapted and interpreted from Verberg and Veldkamp 2001, and Verberg et al. 1999b)



References

Aber, J.D. 1997. “Why Don’t We Believe the Models?” Bulletin of the Ecological Society of America 78(3): 232-233.

Agarwal, C., G.M. Green, J.M. Grove, T. Evans, and C. M. Schweik. *A Review and Assessment of Land-Use Change Models: Dynamics of Space, Time, and Human Choice*. CIPEC Collaborative Report No.1. USFS Publication NE-GTR-297 Joint publication by the Center for the Study of Institutions, Population and Environmental Change at Indiana University-Bloomington and the USDA Forest Service. Burlington, Vt.: USDA Forest Service Northeastern Forest Research Station. Available at http://www.fs.fed.us/ne/newtown_square/publications/technical_reports/pdfs/2002/gtrne297.pdf

Bezroukov, N. 1999a. “Open Source Software as a Special Type of Academic Research (a Critique of Vulgar Raymondism),” *First Monday*, volume 4, number 10, at http://firstmonday.org/issues/issue4_10/bezroukov/, accessed 8 December 2002.

Bezroukov, N. 1999b. “A Second Look at the Cathedral and the Bazaar,” *First Monday*, volume 4, issue 12, at http://firstmonday.org/issues/issue4_12/bezroukov/. Accessed 8 December 2002.

Bonaccorsi, A. and Rossi, C. 2003. “Why Open Source Software Can Succeed.” Forthcoming in Research Policy Special Issue on Open Source Software Development. Available at http://opensource.mit.edu/papers/rp-bonaccorsirossi.pdf.

Burnham, J.C. 1990. “The Evolution of Editorial Peer Review.” *The Journal of the American Medical Association*. 263 (10): 1323-1329. 9 March.

Camara, G. 2003. Open Source GIS Software in Brazil. Presentation at the International Symposium on Open Access and the Public Domain in Digital Data and Information for Science, UNESCO, Paris France. March 11.

CreativeCommons.org. 2004. http://creativecommons.org/license/. Accessed February 17, 2004.

Dafermos, G. N. 2001. *Management and Virtual Decentralized Networks: The Linux Project*. Masters Thesis, Durham Business School, http://opensource.mit.edu/papers/dafermoslinux.pdf, accessed 23 September 2002.

de Koning, G.H.J., Verberg, P.H., Veldkamp, A. and Fresco, L.O. 1999. Multi-scale modelling of land use change dynamics in Ecuador 61 (2). pp. 77-93.

Drucker, P.F., 1999. "Beyond the Information Revolution." *Atlantic Monthly*. Available at http://www.theatlantic.com/issues/99oct/9910drucker.htm. Accessed 22 February 2003.

Feller, J. and B. Fitzgerald. 2002. *Understanding Open Source Software Development*. London: Addison-Wesley.

Fielding, R.Y. 1999. “Shared Leadership in the Apache Project.” *Communications of the ACM*. 42(4): 42-43.

Gasser, L. and Scacchi, W. 2003. Continuous Design of Free/Open Source Software. Preliminary Workshop Report and Research Agenda. Available at http://www.isr.uci.edu/events/ContinuousDesign/Continuous-Design-OSS-prelim-report.pdf. Accessed 18 Feburary 2004.

Glass, R. 1999. "Of Open Source, Linux and Hype." IEE Software 16(1): 45-51

Godfrey, M.W. and Tu, Q. 2000. "Evolution in Open Source Software: A Case Study," at http://plg.uwaterloo.ca/~migod/papers/icsm00.pdf, accessed 25 August 2002.

Ghosh, R.A. 2003. "Cooking Pot Markets: An Economic Model for the Trade of Free Goods and Services on the Internet." *First Monday*. 3(3). Available at http://www.firstmonday.org/issues/issue3_3/ghosh/

Ghosh, R.A., Robles, G. and Glott, R. 2002. *Free/Libre and Open Source Software: Survey and Study*. Technical report. International Institute of Infonomics. University of Maastricht, The Netherlands. June. Available at http://www.infonomics.nl/FLOSS/report/index.htm.

Grove, J. M., C. M. Schweik, T. P. Evans and G. M. Green. 2002. Modeling human-environmental dynamics. Pages 160-188 in *Clarke, K.C., B.E. Parks, and M.P. Crane (eds.)* Geographic information systems and environmental modeling, Prentice Hall, Upper Saddle River, N.J.

Hann, I., Roberts, J. and Slaughter, S. 2002. "Why Do Developers Contribute to Open Source Projects? First Evidence of Economic Incentives."

Hertel, G., Niedner, S. and Herrman, S. 2003. "Motivation of Software Developers in Open Source Projects: An Internet-Based Survey of Contributors to the Linux Kernel." In Feller,J. Fitzgerald, B., Hissam, S. and Lakhani, K. (eds.) Taking Stock of the Bazaar: Proceedings of the 3rd Workshop on Open Source Software Engineering. Available at http://opensource.ucc.ie/icse2003.

Johns, A. 2001. "The Birth of Scientific Reading." *Nature*. 409: 287-289. January 18.

Johnson, J.P. 2002. "Economics of Open Source Software," *Journal of Economics and Management Strategy*. 11(4): 637-662.

Jorgensen, N. 2001. "Putting it All in the Trunk: Incremental Software Development in the FreeBSD Open Source Project." Information Systems Journal. 11(4): 321-336.

Kronick, D. 1990. "Peer Review in 18th Century Scientific Journalism." *The Journal of the American Medical Association.* 263 (10): 1321-1322.

Lakhani, K.R., Wolf, B., Bates, J. and DiBona, C. 2002. "The Boston Consulting Group Hacker Survey, Release 0.73." Available at http://www.osdn.com/bcg/bcg-0.73/BCGHackerSurvey0-73.html. Accessed 9 September, 2002.

Lawrence, S. 2001. Online or Invisible? *Nature* 44 (6837): 521.

Lee, S., Moisa, N., and Weiss, M. 2003. "Open Source as a Signalling Device: An Economic Analysis." In Feller,J. Fitzgerald, B., Hissam, S. and Lakhani, K. (eds.) Taking Stock of the Bazaar: Proceedings of the 3rd Workshop on Open Source Software Engineering. Available at http://opensource.ucc.ie/icse2003.

Lerner, J. and Tirole, J. 2002b. "The Scope of Open Source Licensing." Available at http://opensource.mit.edu/online_papers.php. Accessed July 14, 2003.

Lessing, L. 2001. *The Future of Ideas: The Fate of the Commons in a Connected World*. New York: Random House.

Markus, M.L., Manville, B. and Agres, C.E. 2000. "What Makes a Virtual Organization Work?" *Sloan Management Review*, Fall. Pp. 13-26.

Mockus, A., Fielding, R., and Herbsleb, J. 2000. "A Case Study of Open Source Software Development: The Apache Server." *Proceedings of he 22nd International Conference of Software Engineering*. Pp. 263-279.

Moody, G. 2001. *Rebel Code: Linux and the Open Source Revolution*. Cambridge, MA: Perseus Press.

Opensource.org, 2002. "The Approved Licenses," at http://www.opensource.org/licenses/index.html. Accessed 17 February 2004.

NIPE. 2004. Terralib. Available at http://terralib.dpi.inpe.br/home.html. Accessed 19 February 2004.

Norris, J.S. 2004. Mission-Critical Development with Open Source Software: Lessons Learned. *IEEE Software* Jan/Feb.

Noth, M., Borning, A., Waddell, P. (2003). An extensible, modular architecture for simulating urban development, transportation, and environmental impacts. Computers, Environment and Urban Systems 27(2). pp. 181-203.

Parker, P., Letcher, R., Jakeman, A., Beck, M.B. et al. 2002. Progress in integrated assessment and modeling. Environmental Modelling and Software 17: 209-217.

Pavilicek, R.C. 2000. *Embracing Insanity: Open Source Software Development*. Indianapolis, IN: SAMS Publishing.

Perens, B. 1999. "The Open Source Definition." In C. DiBona, S. Ockman and M. Stone (editors.) *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA: O'Reilly and Associates.

Schweik, C.M. 2004. The open source, open content land-use change modeling effort. Available at www.lulc.org. Accessed Feb 15, 2004.

Schweik, C.M. and Grove, J.M. 2000. "Fostering Open-Source Research Via and World Wide Web System." Public Administration and Management, 5(3). http://www.pamij.com/5_4/5_4_2_opensource.html. Accessed 9 June 2003.

Schweik, C.M., Evans, T. and Grove, J.M. 2003. *Initiating an Open Source\Content Landcover Change Modeling Effort. Report of a workshop held on August 22-23 2003*. Available at http://www.lulc.org/bcworkshop_2003/os_lulc_workshop_report_2003.pd

Schweik, C.M. and Semenov, A. 2003. The institutional design of "open source" programming: Implications for addressing complex public policy and management problems. First Monday 8(1). http://www.firstmonday.org/issues/issue8_1/schweik/.

Shaikh, M. and Cornford, T. 2003. "Version Management Tools: CVS to BK in the Linux Kernel." In J. Feller, B. Fitzgerald, S. Hissam, and K. Lakhani (eds.) Taking Stock of the Bazaar: Proceedings of the 3rd Workshop on Open Source Software Engineering. Available at http://opensource.ucc.ie/icse2003.

Sharma, Sugmaran and Rajgopalan, 2002. "A Framework for Creating Hybrid-Open Source Software Communities*." Information Systems Journal*. 12:7-25.

Sourceforge.net., 2004. Available at www.sourceforge.net. Accessed 17 February 2004.

Stix, G. 2003. "Some Rights Reserved." Scientific American, February 10.

Verberg, P.H., de Koning, G.H.J., Kok, K., Veldkamp, A., and Bouma, J. 1999a. A spatial explicit allocation procedure for modelling the pattern of land use change based upon actual land use. Ecological Modelling 116. pp. 45-61.

Verberg, P.H., Veldkamp, A. and Fresco, L.O. 1999b. Simulation of changes in the spatial pattern of land use in China. *Applied Geography* 19. pp. 211-233.

Verberg, P.H. and Veldkamp, A. 2001. The role of spatially explicit models in land-use change research: A case study for cropping patterns in China. Agriculture, Ecosystems and Environment 85. pp. 177-190.

Verburg, P. H., Soepboer, W., Veldkamp, A., Limpiada, R., Espaldon, M. V. O., Mastura, S.S.A. (2002) Modeling the spatial dynamics of regional land use: The CLUE-S model. *Environmental Management*, 30, 391-405.

Verberg, P.H., Schot, P., Dijst, M., and Veldkamp, A. Forthcoming. Land use change modeling: current practice and research priorities. *GeoJournal*. Available at http://www.geo.ucl.ac.be/LUCC/MODLUC_Course/PDF/T.%20Veldkamp%20(intro).pdf

Veldkamp, A. and Fresco, L.O. (1996a) Clue-CR: an integrated multi-scale model to simulate land use change scenarios in Costa Rica. Ecological Modeling 91. pp. 231-248.

Veldkamp, A. and Fresco, L.O. (1996b) Clue: a conceptual model to study the conversion of land use and its effects. Ecological Modeling 85. pp. 253-270.

Veldkamp, A., and Fresco, L.O. (1997) Reconstructing landuse drivers and their spatial scale dependence for Costa Rica (1973 and 1984). Agriculture Systems 55 (1). pp. 19-43.

Voightmann, M.P. and Coleman, C.P. 2003. "Open Source Methodologies and Mission Critical Software Development." In J. Feller, B. Fitzgerald, S. Hissam, and K. Lakhani (eds.) Taking Stock of the Bazaar: Proceedings of the 3rd Workshop on Open Source Software Engineering. Available at http://opensource.ucc.ie/icse2003.

Waddell, P. (1998) An urban simulation model for integrated policy analysis and planning: residential location and housing market components of urbanism. Paper presented at the 8th World Conference on Transport Research, Antwerp, Belgium. July 12-17. Available at www.urbansim.org.

Waddell, P. (2002). UrbanSim: Modeling urban development for land use, transportation and environmental planning. *Journal of the American Planning Association* 68(3). pp. 297-314.

Waddell, P. and Ulfarsson, G.F. (forthcoming). Introduction to urban simulation: design and development of operational models. In Handbook in Transport, Volume 5: Transport Geography and Spatial Systems, Stopher, Button, Kingsley, Hensher eds. Pergamon Press. Available at www.urbansim.org.

Wichmann, T. 2002. Floss final report part 2. Firm's open source activities: motivations and policy implications. Berlecon Research: Berlin Germany. Available at: http://www.berlecon.de/studien/downloads/200207FLOSS_Activities.pdf

Ziman, J. 1969. "Information, Communication, Knowledge." *Nature*. 224: 318-324. 25 October.