Open Source and Open Content: A Framework for Global Collaboration in Social-Ecological Research

By

Charles M. Schweik
Tom P. Evans
J. Morgan Grove

Preprint manuscript date: 2/22/05

Forthcoming in Ecology and Society (http://www.ecologyandsociety.org/)

Note to readers: This preprint is being published in Ecology and Society, a refereed open access online journal. Consequently, the hyperlinks included are written in HTML code necessary for the preparation of the journal version.
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 resulted in great progress in scientific knowledge over the last four centuries. The process also provides an example of how advances in technology (the printing press coupled with systems for the delivery of mail) can change the speed by which scientific knowledge can grow.

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 costs of scientific publishing that forces the kind of brevity in publication notable in work-count restrictions. A second problem is the cost of the global distribution of paper copies of the journal. A third more recent problem involves the difficulty in making widely available scientific knowledge based on complicated, technical expertise, often grounded upon computing. For example, in the context of land-use change modeling (a topic discussed later in more depth), significant and complex models exist, but there often is slow growth or application beyond the model’s original founding group because of the transaction costs required to learn and apply the model elsewhere. Moreover, Aber (1997:232) notes that in the field of ecology there is a general distrust of such models because these projects and papers are assumed to be 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?

Drucker (1999) observed that when new technologies emerge, it often takes years before real innovation utilizing this technology occurs. Such a lag exists in the area of scholarly communication. For example, with the exception of one or two emerging innovators, most “e-journals” follow the design of the old paper model – volumes and issues, with hyperlinks to papers or abstracts. While this is an important innovation in itself (e.g., making scholarly material more accessible world-wide), web technologies have the capacity to adapt traditional methods of scientific communication and collaboration to systems that are much more powerful, immediate, and interactive. Internet-based technologies and collaborative approaches that have been around for nearly a decade can be applied in new ways to enhance the way scientific research is communicated and built upon. This paper describes some of the features that might be present in a new framework for collaborative scientific research that includes:

(1) The Web as an interactive platform for communicating throughout the research process;
The traditional concept of “peer-review” as a mechanism for maintaining quality control and as a mechanism to encourage participation;

“Open Source” (OS) software licensing and development practices; and

The emerging phenomenon of “Open Content” (OC) licensing.

Taken together, these four components create the foundation for a new approach to global scientific collaboration, in the form of what we refer to as a “next generation peer-reviewed e-journal.” Undoubtedly, readers of *Ecology and Society* are familiar with the Web (#1) and the traditional peer-review process for evaluating scientific research (#2). Therefore, this paper will elaborate on #3 (OS software licensing and collaborative practices) and #4 (OC) in order to describe more fully these components and to raise important issues we see looming on the horizon.

To do this, we first describe the concept of OS software licensing and related aspects of OS collaborative development practices. Second, we describe the emerging OC licensing phenomenon. Third, we address the question some may ask as to whether the OS approach is really a special case of the traditional scientific process. We see several important differences. Fourth, we provide a vision of the future – how we might apply an OS/OC approach to domains beyond software development and into more broadly scientific research. To do this we utilize an example we are thinking about that is directly relevant to *Ecology and Society* readers: employing OS/OC collaborative principles in land-use change modeling. Using this example, we raise several important issues that will need deeper considerations for the proposed collaborative paradigm to become operational.

**OPEN SOURCE SOFTWARE LICENSING AND COLLABORATIVE DEVELOPMENT PRACTICES**

Most software developed by firms or individuals are treated as proprietary with full copyright protection, and is distributed as compiled (unreadable) executable programs. In other words, such software is treated as private or toll goods (Ostrom and Ostrom, 1977), and the proprietary license agreement typically prohibits further copying of the software for use on other computers.

But beginning in 1984, Richard Stallman and others who are now part of the [Free Software Foundation](http://www.fsf.org) (FSF), initiated a social movement in the software industry, arguing that because of the digital properties of software, users should be provided freedom to use, modify and distribute the software as they deem necessary (Stallman, 1999; 2002). Consequently, this type of software is often referred to as Free or Libre (FL) software ([FSF, 2004](http://www.gnu.org/philosophy/free-sw.html); Ghosh et al., 2002) which is different from the commonly used term “freeware.” A major innovation in the effort by Stallman was the principle of “copyleft,” an approach that uses copyright law in a way that maintains the following
freedoms for software users: (1) the right to run the software; (2) the right to read the software source code and modify it; (3) the right to redistribute the original version of the software; and (4) the right to redistribute modified versions of the program (Stallman, 1999). The principle of copyleft also stipulates that all future versions of the software are assigned these same principles (<a href="http://www.gnu.org/licenses/licenses.html#WhatIsCopyleft">FSF, 2004</a>). The FSF summarizes the approach this way:

“To copyleft a program, we first state that it is copyrighted; then we add distribution terms, which are a legal instrument that gives everyone the rights to use, modify, and redistribute the program's code or any program derived from it but only if the distribution terms are unchanged. Thus, the code and the freedoms become legally inseparable." (<a href="http://www.gnu.org/licenses/licenses.html">FSF, 2004</a>)

Stallman implemented these principles through a software license referred to as the GNU General Public License or GPL.

Open source (OS) software licensing follows these same general principles, and is a derivative of Stallman’s Free Software work (Perens, 1999). As Weber (2004: 16) notes: “Property in open source is configured fundamentally around the right to distribute, not the right to exclude.” However, OS licenses do not always follow the complete “freedom standards” set out by FSF, and therefore are seen as a different “political camp” from the FSF. The difference between the OS and FL software has to do with other restrictions provided in these OS licenses that may limit the freedom of users in what they can do with the software (Perens, 1999). For example, some OSS licenses may allow their software to be used in proprietary software packages, a practice that the FSF rejects. More than 50 OS licenses now exist (see <a href="http://www.opensource.org/licenses/">opensource.org</a>), demonstrating that authors of software have an array of rights they can keep or relinquish rather than the simple dichotomous choice of either full copyright (“all rights reserved”, the default condition) or no copyright or “public domain.” (For comparisons of various FL and OS software licenses, see Perens, 1999 or <a href="http://www.gnu.org/licenses/license-list.html">www.gnu.org</a>).

The permissions to copy, modify and distribute readable software source code found in Free/Libre or Open Source software (FOSS) software licenses provides at least two potential advantages over the traditional proprietary, full copyright approach to software development. All FL and most OS software are provided at no (monetized) cost to the end user. This creates a major incentive for it to be utilized, especially by people and organizations that are working under limited budgets (Hahn, 2002). Second, by providing readable source code and allowing new derivative works to be made, FOSS licenses create the potential to generate a large community of users and developers – larger than any one proprietary organization could create – to develop, test and debug future versions of the software (Raymond, 1998).
This innovation in software licensing, coupled with Internet-based collaborative tools, resulted in a new form of collaboration. FOSS projects represent a form of “commons” (Ostrom, 1990; Dietz et al. 2003) but one that differs slightly from the environmental commons that most readers are familiar with (Hardin, 1968; Ostrom, 1990). In FOSS commons, groups of people act collectively to produce a public good (the software), rather than over-appropriate the resource (e.g., Hardin, 1968). And, importantly, FOSS software projects produce this public good through a common property regime (Benkler, 2002; Boyle, 2003). These regimes typically have some type of governance structure with one or more individuals having authority to prohibit others from contributing if they so desire (Schweik and Semenov, 2003; Weber, 2004).

The distributional and “readable source” requirements of most FOSS serve additional social purposes beyond the issue of property rights. Deep social and cultural roles and norms have been the driving force for these licenses. They promote the ideals of sharing, collaboration, learning from the reading other’s work, learning from peer review of one’s own software contributions, and feeling part of a “virtual community” or a social movement (Bollier, 1999; Stallman, 2002). These kinds of social norms drive peoples’ willingness to participate in the production of many FOSS projects, and are important for driving similar collaborations in the context of scientific research. Schweik and Semenov (2003) show that there are many similarities between some of these norms and incentives of OS programmers and those faced by scientists or academics. This has important implications for the proposed extension of OS approaches into scientific research domains that we describe later in this paper.

Thousands of FOSS products are currently under development. At <a href="http://www.sourceforge.net">sourceforge.net</a>, a prominent OS project hosting website, there are nearly 90,000 hosted FOSS projects and over 900,000 registered users (Sourceforge.net, 2004). Many of these projects are surely obsolete. But even if ten percent 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 global user and development community.

But while the “copyleft” licensing principle is a major innovation that has led to the emergence of FOSS, there are other institutional and infrastructural components that drive and support FOSS collaborations. Given the complexity of this topic and the brevity needed for this paper, we cannot address all of the collaborative components here. But we will mention three important ones: (1) a deeply committed and/or funded participant base; (2) an Internet-based collaborative infrastructure; and (3) an established system of project governance. Let us elaborate on each, in sequence.

First, in the early days of FOSS development (and corresponding literature about this phenomenon), the emphasis was on the volunteer nature of the developer communities who contributed to “flagship” FOSS software like the Linux operating system or the Apache Web Server. Indeed, this volunteer nature still drives many FOSS projects. An argument often made to support of the FOSS software
development model is that it has the potential to create teams with participant numbers rivaling or exceeding anything commercial firms can establish – the now famous “with more eyes, all bugs are shallow” quote by Raymond (1998). But more recent studies of FOSS software report that the majority of projects have only a handful of participants (Ghosh, Robles and Glott, 2002; Dempsey et al. 2002; Krishnamurthy, 2002; Healy and Schussman, 2003) and even in some very large projects (measured by numbers of people associated with them) only a small percentage of participants appear to be performing the work (Warsta and Abrahamsson, 2003). But what is clearly demonstrated in some of the real success stories in FOSS, such as the case of Linux and Apache Web Server, is that the early developers were passionate in what they were doing (often driven by a feeling of a social movement such as Stallman’s “freedom” ideas, or the idea to “take on” the dominance of Microsoft products). These social movements motivated some developers to contribute their time and intellectual property without monetary compensation. Today, we are witnessing what appears to be a shift by some firms in the software industry to place human and organizational capital behind the development of FOSS products that are considered strategic interests (e.g., IBM, Hewlett Packard, etc.). And some governments are starting to place more emphasis on the use and possible support for FOSS as well (Hahn, 2002). In short, either a passionate developer community, or some financial support, or both are required in order for a FOSS project to succeed.

Second, regardless of the size of the participant community, a set of Internet-based tools for communication and coordination is required (Kelly and Jones, 2001; Dube, 2001; Halloran et al., 2003; Shaikh and Cornford, 2003). Components include version and concurrency control, build management, workplace management, as well as change request and release management (Asklund and Bendix, 2002). Many of these projects utilize systems like Concurrent Versioning System or CVS, and project management platforms like Sourceforge.net and various communication tools (e.g., chat/instant messaging, discussion forums, Wikis, email, FOSS project web pages, etc.).

Third, as projects become larger in scope and in participants, it is likely that some form of governance structure will be required to make decisions about important project issues (Schweik and Semenov, 2003). Such a governance structure might include: (1) prioritizing features to include in new versions of the software; (2) defining rules and procedures on how production will proceed; and (3) assisting in the resolution of disputes between team members. For example, Sharma, Sugumaran and Rajgopalan (2002:13) note that OS software communities create and abide by sets of rules which are modified over time as the project matures. And studies such as Divitini et al. (2003) and Shaikh and Cornford (2003) provide examples of conflict in OS software settings. This is one area of FOSS projects that is less understood but could be important for understanding why some projects succeed and some others fail.

To summarize, a licensing innovation, coupled with a motivated or financially supported developer community using Internet-based collaborative infrastructure tools
and working within some set of governance principles are the core elements of FOSS software collaboration projects. This configuration provides the foundation for a new paradigm in collaborative scientific research. But making that transition requires attention to the next innovation in FOSS licensing – what we refer to as “Open Content” licensing.

(Note: Readers interested in more information and studies on elements of FOSS, there are at least three relatively large research repositories on the FOSS phenomenon. See, for example, the MIT Free/Open Source Research Community Web site, the Open Source Resources for Researchers and the Open Source Software Research program at the Institute for Software Research at the University of California at Irvine.)

THE EMERGENCE OF OPEN CONTENT LICENSING

Although the principle of “copyleft” and FOSS software licensing has been around for two decades, it is not widely recognized that this licensing innovation is applicable to other intellectual property “content” in addition to computer code (Bollier, 1999; Schweik and Grove, 2000; Stallman, 2001; Weber, 2004).

This idea was first advanced by Stallman’s group as well, who felt the “freedom philosophy” not only needed to apply to software, but also to user guides, technical documentation, etc. In this context, they developed the GNU Free Documentation License, to govern the use, modification and distribution of software documentation. They also set requirements on what parts of the document must remain unmodified from version to version (such as the original author’s copyright notice), the terms of distribution, and a list of previous authors (Stallman, 1999).

Around the year 2001, a new set of licenses following similar principles was developed by people associated with CreativeCommons.org that can be applied to intellectual property created by authors or artists. These developers have created a set of Creative Commons (CC) licenses that allow authors to select, in a modular way, which rights they want to reserve and which to relinquish related to the content they have produced. Key issues and questions, related to the software licensing above, drive various CC license “module” choices including:

1. Distribution. Can readers freely copy and distribute this intellectual property? If yes, the user of the content can download and freely distribute the associated content.

2. Derivative works.
2.1 Can derivative work be made using the content as a base? If yes, the user of the software or content can create new works based on the available and readable digital files.

2.2 If derivative works are permitted (yes to question 2.1) then, must these derivations fall under the same license as the parent work (a “viral” licensing scheme; Pavlicek, 2000) or can they be distributed under a different licensing scheme?

3. Author attribution. 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.

One distinction between Stallman’s GFDL and the CC licenses is that the CC licenses are more modular. This means that authors can select, through the answers to the above questions, which rights to maintain and which rights to give away. With such licensing, authors of other content (e.g., music, artwork, academic or scientific papers) now have the opportunity to hold “some rights reserved” (Stix, 2003), in a fashion similar to what has occurred in the OS software development domain. In this paper, work other than software or software documentation will be referred to as “Open Content” (OC).

Good examples of the use of Stallman’s GFDL can be found at various “wiki” sites such as <a href="http://http://en.wikipedia.org/">Wikipedia</a> and <a href="http://http://en.wikipedia.org/wiki/Wikibooks">Wikibooks</a>. Interested readers may want to review the entry for “open content” in Wikipedia, which provides a list of some emerging open content projects.

IS THE FOSS APPROACH SIMPLY A SPECIAL CASE OF THE TRADITIONAL SCIENTIFIC PROCESS?

Some have suggested that the traditional method of submitting papers to refereed journals, and publishing one’s findings are similar to the FOSS development process (Bezroukov, 1999a; 1999b). But in our view, there are some important differences (Schweik and Semenov, 2003). First, in FOSS collaboration settings, the entire research product (software) is shared with the community, including the “research” (software development) process, rather than just the equivalent of a final paper that summarizes methods and provides end results. This differs from traditional publishing, where space restrictions in journals limit what can be provided to the community. Second, the open access nature of FOSS collaboration over the Internet increases the size of the community that might contribute to the project. Third, the right to freely copy and distribute copies of the intellectual property (software) is very different from the policies of many scientific journals that hold full copyright and require the reader to obtain permission before duplication can be made. Fourth, FOSS collaboration over the Internet potentially increases the speed in which innovations can be published. Systems of peer-review in FOSS contexts might take similar amounts of time compared to that of traditional peer-review processes in scientific journals, but the act of publishing the results (e.g.,
improvements to a program module) can be dramatically increased – in fact, be nearly immediate – after the peer-review process is completed.

In short, the convergence of (1) collaborative principles related to FOSS development, (2) the emerging area of OC licensing, (3) the Web as a platform for scientific communication and (4) the traditional concept of peer-review, provides the foundation for a new paradigm in the production of collaborative scientific research. Weber (2004: 17) articulates this nicely: “The open source process has generalizable characteristics, it is a generic production process and it can and will spread to other kinds of production.” We refer to this emerging paradigm in science as “Open Source and Open Content-based collaborative research or “OS/OC research,” for short. The remainder of this paper focuses on important issues that will need further thought in order to make this transition to OS/OC research. To place these issues in some context, we focus our discussion around one area of research relevant to readers of Ecology and Society: land-use change modeling.

APPLYING AN OS/OC COLLABORATION APPROACH IN SCIENTIFIC RESEARCH: THE EXAMPLE OF LAND-USE CHANGE MODELING

Land-use change modeling is an important area in the domain of ecology and society, and is an area ripe for an OS/OC collaborative endeavor for several reasons. First, there is substantial (global) interest in such tools. Scientists and government officials desperately need such models to help them better understand how land-use systems function, to explore the effects of various policy or planning initiatives and to assist in the building of consensus (Dutton and Kraemer, 1985; King and Kraemer, 1993). Second, in the current system of research and publication, cross-organizational participation in model development is happening at a snail’s pace – if at all.

We base this statement on recent reviews of available land-use change models (U.S. EPA, 2000; Grove et al., 2002; Agarwal, et al., 2002). These reports identify more than 30 different models currently available (some in the public domain, some FOSS licensed, some proprietary packages), each using different approaches and technologies. While having a relatively large number of land-use change models is beneficial in terms of innovation, building upon this prior work is difficult because these models often: (1) require special disciplinary expertise that may not be available; (2) are technologically and theoretically complex; (3) require extensive data; and (4) are not readily available, in part because of the publishing limitations. Consequently, a researcher or policy-maker wishing to try to apply a model to another geographic location faces substantial transaction costs. In short, this model review literature makes it apparent that the scientific community needs to find better ways of building upon each other’s work, for it appears that most of the evolution of individual models over time comes from the same groups that initially developed the models.

Parker and colleagues (2002: 212) raise similar issues in their discussions of 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 different 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/OC 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 this be accomplished?

In August 2003, we held an interdisciplinary workshop to initiate such an effort (Schweik, Evans and Grove, 2003). Several land-use change models were represented and participants came from a variety of academic institutions and organizations active or with an interest in land-use change modeling. UrbanSim (Waddell 1998; 2000) – a simulation model for integrated planning and analysis of urban development – was one such model that garnered particular interest from workshop members, and it is the one we use to better describe the issues of OS/OC modeling we see ahead. (Interested readers can find a short summary of UrbanSim in Appendix 1). Technically, it has a “model coordinator” program and five sub-modules: Accessibility, Economic and Demographic Transition, Household and Employment Mobility, Household and Employment Location, and Real Estate Development.

As an example, UrbanSim demonstrates the multi-disciplinary nature and level of sophistication (technically, theoretically, empirically) involved in current-day land-use change modeling (see U.S. EPA, 2000 and Agarwal et al., 2002 for descriptions of other land-use change models). Applying the model involves data gathering, formatting, and a variety of steps that need to be taken (some manually by the analyst, some through automation such as the Model Coordinator – see Appendix 1). And, as in FOSS development situations, in order to make some improvements or alternations to a sub-module, some programming skills may be required. But from the standpoint of making scientific advances in the area of land-use change modeling, moving to an OS/OC 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 and ecology, to name a few) who are 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, different driving variables of change and processes will become important as these models are applied in various regions of the Earth. If applications of models like UrbanSim expand and are applied to other geographic areas (which they are), it is likely that the logic within various sub-modules would need modification in order to capture the effect of different relevant processes. For example, we would expect some variation in the causes or drivers of change (Lambin, Geist and Lepers, 2003) when comparing the contexts of Phoenix, Arizona, USA, and Baltimore, Maryland, USA.

There are many ways that individuals could contribute to an OS/OC project in general and more specifically to the further development of land-use change models like
UrbanSim. People with the technical skills to “work under the hood” of a model (e.g., researchers with programming skills) could write or modify existing code or contribute entirely new modules, thereby extending the operations, manipulations, or analyses that data sets might be subjected to. Others with some programming knowledge but with less interest in writing new code or functions could serve a critical role in code review (a form of peer-review in a software context) or in code documentation.

Scientists who lack programming skills could participate in the identification of bugs in a model module through its use and testing. With complex modeling frameworks model users outside the core model development team could also make critical contributions in terms of developing user manuals or other modeling documentation, or in conceptualizing new functions – such as model validation approaches and tools (Pontius, Huffaker and Denman, 2004) – for someone else to program. In addition, new model users (and developers) can facilitate the “many eyes” approach (Raymond, 1998) where new ideas and insights may be contributed to the original model that its developers would not have considered. A critical aspect of this approach would be to provide enough distance learning material to allow the model to be explored and applied by researchers from disciplines other than those represented in the model’s development. And importantly, similar collaborations could involve users developing “how to” documentation, or theorists providing key insights though papers placed under some form of OC licensing. Last, but perhaps most importantly, theoreticians who are not modelers can still contribute crucial elements to the collaborative endeavor, by contributing the theoretical base through papers that provide the foundation for the logic of particular model components.

From an empirical standpoint, an OS/OC licensing approach would encourage the application of a particular model like UrbanSim in different geographic areas. Different users or interested stakeholders could contribute data (perhaps also under an OC license) that can be used to test a single model in two different locations that have different social-ecological situations. These comparative applications would likely result in new specifications for model refinements. Models licensed as “new derivative works permitted,” could then be tailored accordingly by the same researcher or another in the collaborative community, leading to insights that would not have been apparent by running the model only in a single location. Under OS/OC, researchers could collaborate in the parameterization of models for different locations. For example, Clarke (2004) recently reported calibration results of the SLEUTH land use change model applied to eleven different cities. Here, five different parameters (e.g., dispersion, breed, spread, slope and road gravity) that drive model growth were reported for each study, building what Clarke referred to as the first inventory of the “DNA” of urban landuse change. The development of these kinds of parameter libraries for various landuse change settings (e.g., urban, rural, forest, agriculture, etc.) at different spatial and temporal scales would be particularly useful mechanisms for cross-site analysis, model testing and identification of proximate and underlying drivers of change (Geist and Lambin, 2002).

This describes the vision we are working toward, and as we have said earlier, this vision is applicable to many areas of scientific research, not just land-use modeling. We
think this is appropriate to all kinds of scientific research, from basic research where papers and analytic approaches are licensed as OC and shared, to more applied research such as in this landuse modeling context. Perhaps a reviewer of a previous version of this manuscript said it best: “… OS and OC techniques when brought together within a project infrastructure,” (discussed below) “may potentially stimulate the advance of science and scientific knowledge though a more open experimentation, innovation, evaluation, versioning, and e-journal publishing regime.” We now turn to a discussion of some critical issues that will need to be addressed to make such a vision viable.

Initiating OS/OC collaboration (Issue 1)

A central first issue is how to initiate an OS/OC collaborative endeavor in scientific research, such as what we are trying to do in the context of land-use change modeling. The question here is: just how do FOSS collaborations start? And what does this mean for OS/OC-based collaborative science?

FOSS projects tend to (but not always) follow a three-stage trajectory (Figure 1) (Schweik and Semenov, 2003). They often are initiated by a small group of one or more people who possess programming skills and a common (software) need (Nakakoji, et al., 2002). These small groups work to develop a core piece of software, which may not even at this point be licensed FOSS (Figure 1, Stage 1). At some point the team member(s) decide the software is ready to “go open” and assign the software a FL or OS license and make it available over the Internet (Stage 2 in Figure 1). Many projects hope that soon after that they might achieve “high growth” (Figure 1, Stage 3) in terms of gaining a larger development community and a sizable user community and this is one measure of project success (Crowston et al., 2003, 2004; Stewart, 2004). But projects that continue to be vibrant (in terms of development) with small groups of participants should also be considered cases of success (Figure 1, Stage 3, Small group, Stable Use). In fact, many studies now report that the majority of OS projects never reach high growth and involve small numbers of individuals (Ghosh, R.A. and Prakash, 2000; Ghosh, Robles and Glott, 2002; Dempsey et al. 2002; Krishnamurthy, 2002; Healy and Schussman, 2003; Capiluppi, Lago and Morisio; 2003).

We should note that in recent years, large groups or organizations with already developed close-source software have decided to re-license and make it open (IBM is perhaps the best example of this). Consequently, not all FOSS projects follow the trajectory in Figure 1. But regardless of how they get started, in each stage shown in Figure 1, the project could result in a loss of momentum or interest, and an eventual loss of participation and, ultimately, project death. In sum, the success of a FOSS project is critically dependent on some community of practitioners (developers and users), whatever the size, to ensure that the project continues and evolves over time (Nakakoji, et al., 2002).

What does this mean for the idea of OS/OC scientific collaboration projects? It is likely that these kinds of projects will follow the same trajectories shown in
Figure 1. Many will begin with a small virtual team of interested collaborators, and perhaps remain that way throughout the project’s lifetime. Or, they could generate a larger following or participation base over time. One emerging example that is closer to an OS software example, but exhibits elements of OC and scientific collaboration is the R-Project for Statistical Computing, which appears to be growing in popularity.

We expect that an OS/OC scientific collaboration project such as in the area of landuse change modeling will have a greater chance of achieving a “high-growth” type success if it is able to attract a “community of practitioners” (including scientists, model developers, policy makers, concerned citizens, etc.) who are online. Consequently, we would hypothesize that projects that are sponsored or sanctioned by a well-established scientific association will have an edge over projects that do not. Projects sponsored by a well known and respected association will be seen by other potential participants as more legitimate and worthy of their time.

In our specific case of trying to initiate such collaboration around the land-use change model UrbanSim, we recently proposed these ideas at the first ever held workshop for model users (UrbanSim, 2005). In addition to the UrbanSim model development team, in attendance at this workshop were policy-makers and modeling technicians from a variety of metropolitan planning organizations all trying to apply the model to help them make regional planning decisions related to transportation infrastructure, smart growth, etc., as well as interested parties from several U.S. universities. When asked, most participants were receptive to the idea of OS/OC collaboration. Clearly, attention to incentives for participation are key (see next section), but we left the meeting optimistic that people would be willing to share their expertise through the act of contributing some documentation, as long as someone else would write up their experiences in an area of the modeling effort that they have yet to face.

Incentives for Participation and E-journal Publishing (Issue 2)

Perhaps the biggest puzzle in recent years about FOSS collaboration is the question of why volunteer programmers donate their time and intellectual property (Glass, 1999; Lee, Moisa and Weiss, 2003; Ghosh, 2003). Recent studies note some motivations, which include: meeting some personal software need; addressing a software crisis; intellectual stimulation; the desire to belong in some community; contributing to the free software movement; or simple altruism (Lakhani, et al. 2002; Feller and Fitzgerald, 2002; Hertel, Niedner and Herrman, 2003). Human capital and signaling theory provides an economic explanation: participation in OS projects allows actors to learn from others’ source code and receive feedback from others on their own contributions (Hann et al., 2002; Voightmann and Coleman, 2003). In this regard, contributions are seen as future-oriented investments in building one’s career (Lerner and Tirole, 2002; Johnson, 2002; Lee, Moisa and Weiss, 2003). Moreover, high quality contributions act as a signaling device of a participant's programming abilities (Lee, Mosia and Weiss, 2003) and help to establish a reputation (Sharma, Sugmaran and
Rajgopalan, 2002), which could lead to future work opportunities (Lee, Moisa and Weiss, 2003).

But in some cases of FOSS software, teams are not comprised entirely of volunteers. In one study, nearly one-third of the FOSS developers surveyed are directly paid by employers to participate (Ghosh et al. 2002). Wichmann’s (2002) study of 25 firms active in Linux development found that most 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 FOSS products.

What does this mean for applying the OS/OC collaborative paradigm to other scientific domains such as land-use change modeling? The motivations between FOSS programmers and scientific researchers are not all that different (Schweik and Semenov, 2003). Many in academia (graduate students, junior or even senior faculty) would be willing to hone new skills through the distance-learning components of reading “source” (e.g., models, papers, etc.) and peer-review with feedback. The motivation to “signal” ones abilities is also important particularly for junior scientists looking for jobs or to gain prestige. As one scholar put it during our workshop (Schweik, Evans and Grove, 2003): “Had I known about it, I would have gladly licensed my model OS in graduate school. That way others might have used it and it would have gotten my name more widely known.”

But in order to signal one’s abilities by posting intellectual property, one’s name needs to be associated, over time, with that submission. We predict that a key motivation in OS/OC projects will be the assurance that the intellectual contributions to the broader project by individual participants will be tracked and archived, over time. In the context of FOSS code, this type of tracking is mandated as part of the licensing agreement. In the GPL, for example, authors of a completely new program place a copyright notification comment as original authors of a work at the start of the source code with a pointer to the location where the full notice can be found. The GPL requires this copyright statement to stay with all future derivative works based on that original code. It also requires that authors of new derivative work update the software comments with a prominent notice that changes were made, who they are, and the date of the change (<a href="http://www.fsf.org/copyleft/gpl.html">FSF, 2004</a>). In the OS/OC context, scientists will be more likely to contribute new research content if they are able to maintain copyright over their original submissions in the same way a programmer does using the GPL. Mechanisms will need to be developed to attach similar copyright information to any research product (e.g., a paper, a dataset, an analytic module of some sort). This is easily accomplished in content in the form of documents, and could be done by specifying the copyright information or update histories in metadata documentation for other components like datasets. In short, the infrastructure built to support an OS/OC collaboration in science (discussed more below) will need to include a good historical record of how someone contributed over time to a new model module, to a new derivative paper on land-use change theory, to empirical findings, or to other project content.
Another aspect related to the incentives of scientists and academics to participate in OS/OC projects is a consideration of the demands and expectations of their employers. For many scientists, having their work published in high quality, refereed journals is a key measure of success in their field and an important measure 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 OS endeavor. Consequently, we believe for any viable OS/OC scientific collaboration to succeed, it must be placed in the context of current evaluation systems at universities and scientific research organizations. This means publishing in this context would require not just a collaborative web site, but also a “next generation e-journal” component that follows the traditional peer-review process.

We refer to this as a “next generation e-journal” because it would not only publish traditional peer-reviewed “final” content (e.g., papers on theory related to land-use change or the results of empirical studies applying a particular model), but it could also publish work in progress in the form of working papers, new or revised versions of complete models or sub-component modules, and new distance learning material related to the model. In some instances, even datasets used by models 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, broadly defined. This is vastly different than the maximum 30 or so pages that most journals (even most e-journals) currently accept.

Admittedly, this idea is radical and one that is difficult to impose on the current scientific culture. It means a change in what we consider what is publishable. But recently, both computer scientists and librarians have recognized that this change is needed; there is subtle evidence toward a shift from publication as product to publication as process (Lougee, 2004). For example, Berghel (2001: 18) sees a future where the phrase “digital library” will refer to “articulated processes and procedures” rather than a “dull and lifeless era of publishing… of ‘things.’”

We should note that traditionally researchers tend not to share work until it is fairly complete, and then first as a working paper to a more limited audience (such as in a conference presentation) for the purpose of soliciting feedback before submission to a journal for publication. How might this process work in the next generation e-journal we propose? And how could such a system help to protect the intellectual property of an author given that it is being offered publicly but not yet in a final, more traditional published form? This question involves both incentives to participate (the current issue being discussed) and aspects related to project infrastructure (Issue 3) that follows this section.

One possible approach is being discussed by scientists associated with Boston College’s &lt;a href="http://www.urbaneco.org">Urban Ecology Institute</a> and the U.S.D.A. Forest Service as they work to develop the “Urban Ecology Collaborative” (UEC). This collaborative would involve people from a variety of governmental agencies
and universities studying Urban Ecology in a number of U.S. cities. One can think of the UEC idea as a kind of professional research association focused on a common research area or agenda. In this case it is around Urban Ecology, but a similar idea could be developed around Land-Use Modeling, Ecology and Society, or any research program however broadly defined. In ideal cases, the research program would be associated with a formal professional organization which already publishes an associated scientific journal and this idea would extend that journal.

The idea proposed at UEC meetings is that the Internet collaboration infrastructure could be designed to serve three levels or “tiers” of publishing. The first tier would be a “Gallery” that would contain high quality (peer-reviewed) information about the broad research program, in this case Urban Ecology. The Gallery would summarize the program, make available educational materials and would provide information on breaking news and events in the field. In some instances, this could be analogous to a website of an established professional association. Content on these pages would likely fall under full copyright protection.

A second parallel tier within the UEC would be the formal or official e-journal (such as The Journal of “Urban Ecology” or “Land-use Modeling”), where only refereed components would be published after going through a traditional review process. This would be accessible from the main Gallery page. However, if UEC decided to take an OS/OC approach as proposed here, what is published could be more broadly defined, and include not only final drafts of theoretical or empirical papers, for example, but in the case of land-use modeling, new (and peer-reviewed) versions of model modules, new datasets, distance learning materials or other analytical products that are not, in today’s environment, typically considered peer-reviewed publications. It would be up to the UEC governance body (discussed in Issue 4 below) to determine the appropriate OS/OC licensing options to utilize for each type of published material.

Also connected to the main Gallery and e-journal pages would be the third tier of the collaborative system – probably an Intranet system – that UEC associates refer to as the “Virtual Lab.” This Intranet would be accessible only to researchers or organizations more formally connected to the professional association or research collaborative who are actively working on program-related projects. Mechanisms could be included within the virtual laboratory for posting of draft research products, and would allow the solicitation of feedback from peers, but within a smaller, but still public, community. Like the e-journal tier, the governance body would need to consider appropriate OC licensing options, but it may be at this level that the licensing promotes the most freedom (e.g., new derivative works are permitted). Models of such a facility (although not Intranets) are the <a href="http://www.arxiv.org">ArXiv.org</a> preprint server in Physics or the <a href="http://opensource.mit.edu">opensource.mit.edu</a> preprint server for papers studying open source software issues, or working paper series that are commonplace in many research organizations. But like the e-journal tier, this system would allow for preprints of other “working content” (e.g., data, model modules) as well, and would have other Internet-based communication mechanisms (e.g., threaded discussion lists). And here is the point: In some respects, this Virtual Lab tier would be an online system that
would mimic some of the early-information sharing and dialog that occurs in regular in-person conferences. Technically, it would be fairly trivial to provide an automatic email to subscribers based on interest keywords about new working papers (and at the e-journal tier as well) that have been recently posted, much as formal publishers have already done. Such a notification system and a pre-print or working content-like archive would provide an incentive for the early pre-publishing of work in progress (thus speeding up the research process). At the same time it would protect the author’s personal interests in being known as the generator of the ideas posted to this working content server.

**Project Infrastructure (Issue 3)**

A third important issue is the development of project infrastructure to support OS/OC group collaboration which we have already touched upon in the UEC discussion above. But in addition to ideas such as the tiers described above, there are other critical Internet-based components that will comprise a next generation e-journal to support OS/OC collaboration.

In FOSS projects, group collaboration is supported through web-based communication and version control systems. For example, the OS project management website [www.sourceforge.net](http://www.sourceforge.net) provides group communication functions and software version control systems based on the “ Concurrent Versioning System” or CVS (Fogel, 1999). CVS and other version control systems:

1) provide parallelized read/write access with password protection;
2) archive versions of software;
3) allow for the retrieval of modules;
4) allow for new submissions and protect against the problem of overwriting and errantly eliminating other’s work;
5) document changes and change comments over time (author tracking);
6) provide analysis functions to identify differences between module versions;
7) email subscribers when project components are moved, updated or deleted.

A similar set of functions (along with at least two other functions, described below) would be required to support the land-use change modeling effort, and these functions would need to support a broader set of submission types.

For example, consider a hypothetical case using the UrbanSim model as an example (Appendix A). Researcher A might be interested in applying UrbanSim to a location at the fringe of the urban/agriculture interface. To begin, she may first decide to read an existing OC-licensed paper on the theoretical drivers used in UrbanSim for a western U.S. application. She realizes that some component of the logic does not entirely apply to an east-coast situation. Consequently, she might decide to download the text of this paper, and, assuming that the license attached to the paper permits new derivative works to be made, revise it to fit her case’s context. The OC license could be “viral,” meaning that the new derivative work is required to follow the same license of its “parent” document. It could require her to submit this new derivative paper back to the
project for peer-review. Submission of the paper would invoke a function similar to the posting of enhancements of software (number 4 above) but for papers or “documentation.” The paper would then undergo a peer-review and 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 employed in the theoretical paper and implement a new version of the “household location module” of UrbanSim (Appendix 1, Figure 2). Assuming that the module falls under an OS license such as the GPL, once her modifications are complete, she would be required to submit the new derivative module back to the project for peer-review. Upon a positive review by the project “editorial board”, 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 model might then select between the “West Coast Household Location module” or the “East-Coast Household Location module.”

Critical to making the collaborative ideas proposed here work will be special attention to some kind of author contribution tracking mechanism for all types of contributions – papers (theoretical, empirical), modules, and other project documentation, including distance-learning materials. We mentioned earlier that in FOSS projects, the licensing requires some historical log of author contributions in the main header (comment area) of the computer program or in some kind of credits file (Tuomi, 2004). But in the vision of the next generation e-journal we are describing, even more sophisticated methods of author contribution tracking will be needed because of the vital role such tracking plays in encouraging scientist participation. Potentially new ways for authors to “cite” their contributions will need to be devised, and given the speed in which innovations could be published, citations will be more like the way web pages are cited currently (i.e., journal, year, issue, number, date and link).

A reader might raise the important concern that some, particularly junior scientists, may be hesitant to publish ideas that are in more of a “work in progress” stage to a more public site for fear of being “scooped” by someone else in publishing a more formal, peer-reviewed paper on the ideas. But new systems for tracking “idea submissions” into the “pre-print” or “virtual lab” section of the e-journal, will be required to help document early (in the publication process) contributions made by project participants to give them some level of protection to the ownership of these ideas. In this context, contributions in the form of smaller “idea modules” could be attributed to programmers or authors of software documentation that could be referenced by them in, for example, their Curriculum Vitae.

Another concern with an OS/OC approach to scientific research might be a blurring or clouding of authorship, which could be an especially important issue to junior scientists. Of course in many instances peer-reviewed publication of a paper, or a new version of a module, will involve a process similar to that done currently. Some negotiation between contributors will perhaps need to be made to decide how their names will be listed on the submission. In instances where responsibility for a contribution is even less clear, there are other ways for contributors to receive credit that enhances their
personal job situation. For example, in many academic contexts, external letters of support are also vital for promotion, and this type of activity in “clouded” collaboration could bring an active and important contributor some valuable exposure in the scientific community. In addition, what we are suggesting is that by documenting intellectual contributions through some kind of “history log” of all work products, scientists could document in their CV not only papers they have published in the formal peer-reviewed e-journal but also contributions in other collaborative parts of the project as part of service to the discipline. This suggests a potential change to the culture of what and how we document our intellectual contributions and we expect these changes will be gradual and not easily accomplished, in terms of being as acceptable as more traditional forms of publishing. But in the meantime, junior (academic) scientists can treat these contributions as a form of academic service and cite them as such.

Existing systems of scientific research tend to rely on bibliographic measures such as the number of times a paper is cited, and where it is cited in, as surrogates of the importance of research contributions. The infrastructure we are proposing also provides opportunities 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 provide different functions for the land-use change model they are embedded within. Over time, Module A gains wide interest from other modelers or stakeholders using UrbanSim, and several new theoretical papers on logic or derivatives of this sub-module are submitted. 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 a more important contribution because of the activity, interest and derivative works that followed. Quantitative measures could be generated by the system that would measure how many derivative works were spawned by a submission, and these numbers could be added to the author’s CV as another measure of intellectual contribution.

And still another method of measuring the importance of the contribution that could be applied to all types of research content would be the addition of a voting or recommendation function attached to the content. This follows the lines of the ranking system used in Amazon.com, where researchers who have read a paper or reviewed a model’s logic could then submit their comments and give it an importance ranking (e.g., 1-5 stars following the Amazon.com model). Authors of the contribution could then contact those who have offered comments or criticism to hold further private discussion, initiate future collaboration or they might prefer to engage in public debate and discourse through some communication function the e-journal infrastructure could provide.

In addition to the functions listed in the earlier discussion of CVS, there is the need for at least two other major components related to project infrastructure that are not prominent in FOSS development projects: metadata and data management functions, and distance-learning components. Regarding the former, in the proposed application of land-use change modeling, data will be a primary input to the modeling effort and data availability or access is often a major hurdle to developing a model for a particular geographic area. However, in some countries, standards for documenting data,
particularly geographic data, now exist (see for example the U.S. National Spatial Data Infrastructure guidelines) and there are efforts to develop data clearinghouse servers (see www.fgdc.gov/clearinghouse/clearinghouse.html) to share such data across organizations. Other groups are working on standards for developing “web services” which would allow for real-time access and sharing of geographic data over the Internet (see opengeospatial.org). In the context of projects like land-use modeling, some kinds of data, such as raster land cover data, may be too large to serve over the Internet efficiently, but at least their metadata could be, and could include information on where these data reside, how they can be acquired, and information on their OC license.

Another important component of the next generation e-journal infrastructure requiring special attention and prominent display is distance learning. This feature can help recruit and build new membership. In the context of land-use change modeling, learning 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 material could be licensed OC permitting future improvements (subject to peer-review) by anyone wishing to contribute. And the e-journal might include, as another incentive for participation, a prominently displayed list of “key enhancements needed” that might serve as a place where, for example, graduate students could find potential thesis topics.

To conclude this section on infrastructure, we should note that along with software version control systems like CVS, there are other more recent (and FOSS licensed) tools that provide some of the collaborative functions required for this next generation e-journal we are proposing. For example, the “bugzilla” software provides functions for the reporting and the tracking of software bugs. And other software exist that provide document management capabilities. The DSpace software (Smith, et al., 2003) is a web-based repository system to support and archive digital research and educational material produced by faculty and staff at research universities or other research organizations. There are also web-based portal and content management systems such as Plone as well as other Internet collaboration support software such as Wikis. Moreover, other online systems for e-journal publication and editorial management, like the systems used by this journal, are helping to define the infrastructure needed to support the next generation e-journal we describe.

While none of these have all the functions required (especially in terms of relating metadata such as OC licensing to various research products such as data, model modules, papers, etc.), there are enough tools available that could be possibly configured with modest support to establish the core infrastructure to support this next generation e-journal. There is the need (and an opportunity) for some organization or group of inspired
individuals with programming skills to take the development of this next generation e-journal infrastructure on, and then perhaps be willing to provide their solution to the world via a FOSS license.

Project institutional design and governance (Issue 4)

Because FOSS projects involve common property regimes, project governance is an important issue but also an area that has not been deeply researched (Schweik and Semenov, 2003; Weber, 2004). Ghosh, Robles and Glott (2002) report that the majority of OS software projects they studied are directed by a single “lead developer” who maintains a centralized decision making structure. For instance, studies on the OS Linux operating system describe the lead developer as being a “benevolent dictator,” working with a team of “trusted lieutenants” who are experts in a particular domain (Shaikh and Cornford, 2003; Moody, 2001; Sharma, Sugumaran and Rajgopalan, 2002). In some OS projects, would-be developers work their way into becoming “trusted lieutenants” by first working at the periphery of the project (e.g., offering bug reports) and then progressively contributing more in terms of time, effort, and source code contributions. If their contributions are deemed to be high quality or significant, these developers may move toward the core of the development team bringing with it more say in project governance decisions. Other FOSS projects have a different management structure. For example, Jorgensen’s (2001) study of the FreeBSD OS project identifies a nine-member team heading the project. Members are from and elected by the pool of 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 FOSS projects (Sharma, Sugumaran and Rajgopalan, 2002). Most projects have some sort of established conventions or norms of behavior 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 FOSS collaboration paradigm to collaboration in scientific research will be designing the decision-making structure and system of operational rules and dispute resolution procedures for the virtual team. It is conceivable that the governance structure may ultimately become a combination of how journals today are run and organized (e.g., an editor and editorial board) and how FOSS software projects are organized.

It is interesting to note that the trajectory toward membership to some governing board in a scientific research context will mirror in many ways what was described earlier in the context of FOSS projects. Researchers will start at the periphery, as undergraduates, then graduate students and student researchers, then post-docs, junior faculty and ultimately senior faculty or researchers surrounded by the preceding others.
This is similar to the trajectory described above in the context of FOSS developers. But what may be different is the time scales for which an appointment to a governance position may occur. In a FOSS setting, it may take only a year or two for a gifted programmer to demonstrate his or her skills and knowledge to the broader group, whereas in a scientific context it might take a person a decade or more to achieve a similar status within the research community.

A related institutional design issue that will need to be addressed early by the initial governance team is the kind of OS/OC licensing to be used. Each type of project output (e.g., model modules, model usage documentation, empirical papers, theoretical papers, and datasets) will need to be assigned some OS or OC license. For example, we are advocating that whatever form or technology the land-use model utilizes (e.g., a computer program, a statistical script, etc.) they should be placed under some form of OS/OC license that allows the free copying of the model, requires the model “source” to be readable, and permits the development of new derivative works based on the model structure. In some instances, the model developers might decide to make all related products (e.g., the model modules, their documentation, data, and even theoretical papers) fall under these conditions. However, there will be situations where more restrictive licensing is warranted. For example, we expect that most empirical papers on a particular model application should probably be licensed with a “no derivative work” component, because these types of papers report findings from a particular study at a particular time.

OC licensing of theoretical papers presents a particularly interesting and potentially difficult problem. Recall the earlier example of a researcher writing a new derivative theoretical paper related to household location decisions in the eastern U.S based on an earlier paper by someone else on the theoretical drivers in a western U.S. context. The result would be two separate theoretical papers, the east-coast version and the original west-coast version. This differs from the traditional approach to publishing research, because the second version of the paper may have substantial sections of text taken verbatim from the first paper, with new text added. (This is similar, of course, to what might occur in a software documentation update situation under the GNU Free Documentation License described earlier.)

Readers will quickly note that this treads dangerously close to the issue of plagiarism. And if the licensing for the research paper permits new derivative works, the situation exists where someone could download a paper, revise it only slightly and in a trivial manner, and then put their name on it as an additional author. Regarding this situation, one reviewer to an earlier draft of this paper remarked (paraphrasing): “The potential complexity of sorting out the degree of a unique contribution could be overwhelming as this moves from this easy case to more substantial changes that require subjective interpretation to assess…. This will require a robust solution if e-publishing is to co-exist with OS/OC licensing.”

While we cannot profess to have all the answers to this issue, we do have some ideas on how to potentially address it. A conservative approach might be to lean to the
use of the no-derived work licensing for academic papers in general, but still promote other licensing options such as the free copying and distribution, which through simple Internet-based open access will yield more easy access worldwide and will likely lead to a more rapid evolution of the field. Other project output, like distance learning documentation, might be licensed with “derivative works OK” to promote more rapid improvements to learning materials.

But in situations where projects want to take greater advantage of the innovation that could occur in a “derivative works OK” licensing environment, there are several additional options to manage and communicate the degree of contribution made by certain authors. First, as we have stated earlier in the “Infrastructure” section, version control and author tracking systems will need to be devised in next generation e-journal platforms to ensure that the original and subsequent author’s intellectual contributions are tracked over time. As we’ve discussed earlier, to a substantial degree, these issues already exist and have been dealt with in the software and documentation setting of FOSS projects (e.g., history logs). But further empirical work is needed to investigate whether and how this issue has been addressed in the software and documentation contexts.

Second, version control systems like CVS store not only “change logs” of how OS software has been modified from version to version along with who made these changes, but they also archive the past versions of the software modules themselves. The same could be done in an OC setting where the project infrastructure keeps previous versions of other project content, such as theoretical papers. If this was done, it would be a fairly trivial exercise for someone to review the changes made between two versions to assess the real intellectual contribution made by a subsequent author. In fact, commands like the CVS “diff” (difference) command allows one to identify where the differences lie when comparing two different versions of the same content module.

Project finance (Issue 5)

The last issue we will raise relates to the financing of such OS/OC-based research collaborations. Skeptics of the volunteer model of FOSS question the viability of the model over the long term given there is no financial support. For example, this question was raised at a meeting in UNESCO Paris on science and public domain issues (Esanu and Uhlir, 2004), when a participant argued that the only way OS software projects will succeed is if a “government” supports the endeavor. While this is an empirical question yet to be answered, it is highly likely that (at least for major projects) some level of financing will be required. In fact, in the OS software domain, projects can be found that fall under a variety of different financial support schemes, including: (1) the government funding or subsidy model (Hahn, 2002); (2) philanthropic funding such as efforts undertaken by the Andrew W. Mellon foundation; (3) corporate consortia (Hildebrand, 2004); (4) corporate investment (Webb, 2004); (5) venture capital/investment banking; or (6) a hybrid/mix of these. The argument could be made that the long-term success of some of these enterprise OS software projects (even Linux perhaps?) is a consequence of the investments firms like IBM, Sun Microsystems and others are making in such projects (Ghosh et al. 2002, Weichmann, 2002). However, it remains an open question as
to when such funding is absolutely needed for a project to survive versus when an OS software project can exist solely through a volunteer base.

The same question can be raised when one considers extending the FOSS development paradigm into the area of scientific collaboration. However, in the context of academic contributions, it seems that some of the “industry support” may already exist. Universities expect their faculty to undertake research that contributes to a larger research program. Service to a broader professional community is seen as a positive thing in most faculty annual evaluations. If a faculty member wanted to participate in a virtual OS software collaboration as part of his or her research program, most universities would be supportive of this so long as the individual continued to meet traditional measures of scholarship and productivity (such as publications in refereed outlets). This incentive raises again the importance of closely linking or merging the OS/OC research collaboration to a refereed e-journal, for it is likely it would greatly enhance the number of possible researchers willing to contribute.

This leads to the question of how such an e-journal might be financed. Traditionally, scientific journals tend to be published either by academic or professional societies or by commercial publishers. A study recently commissioned by the Wellcome Trust (SQW Limited, 2004) estimates the cost to be about $2,750 per article in good-to-high quality subscriber-pays journals and about $1,425 per article in medium-quality journals. This covers activities such as managing the referee process, editorial support, subscription management, sales and marketing, and final printing costs. Author pays journals tend to be lower, in the range of $1950 for good-to-high quality journals and $1,025 for medium quality journals (SQW Limited, 2004: 3). This same study assumes that the cost of electronic publishing is about the same, given that the cost of maintaining the electronic system replaces conventional distribution costs. Our idea of moving to a web-based e-journal that includes some level of referee process for not only theoretical and empirical papers but also models and model documentation, data, distance learning documents will probably raise the level of time required by reviewers and various component editors and raise the costs of these activities.

It is an open question whether traditional journal publishers following a traditional “user-pays” subscription business and taking an “all rights reserved” philosophy would embrace such an idea. Commercial journals tend to provide e-journal access as a companion to the already established printed version or as part of a larger package of journal subscriptions provided in bulk to large institutional subscriptions (e.g., research libraries). However, other financing models are being explored to promote more open access to scientific information, such as the “author-pays” to publish model with open access to the journal and the model where academic institutions or research libraries are taking on the publishing of disciplinary e-journals as part of their mission (Shortliffe, 2004; Suber, 2004). In this regard, an interesting question requiring further research is whether any components of the business models in the area of OS software (e.g., Weichmann, 2002) are at all applicable and transferable to the process of publishing scientific journals. IBM, for example, has managed to turn OSS into a profitable venture. Could some of what drives their profitability be transferred to the publishing domain? For
example, some of IBM’s profits undoubtedly come from service and support of OS software. Could a possible model based on this idea be for a university to set aside some funds received through a distance learning program to help support a next generation e-journal published by its library and edited by some of its faculty?

In sum, we cannot admit to having a solution to the financing issue, other than to say that experiments in alternative e-journal funding to promote open access are emerging, and a sizable number of FOSS packages are available with some of the needed functionality to support such endeavors. Support from professional societies, foundations, or government agencies (e.g., the U.S. NSF) could potentially invest in supporting the development of a next generation e-journal infrastructure and provide this infrastructure under a FOSS license for other groups to use. Other financing will surely be needed to support such e-journals, but possibly using the “user pays” model or in the case of specialized research areas, perhaps through support as part of a research library mission at a particular institution. Whatever approach is taken, clearly, financing is an issue that will need to be carefully considered in order for the ideas here to work.

**CONCLUDING REMARKS**

A primary innovation in FOSS is the development of new forms of copyright through alternative licensing structures. Importantly these licenses identify the rights to distribute the software, rather than solely focusing on the right to exclude people from using the software (Weber, 2004). Another important innovation is how collaborative development groups operate and are governed, which to a large degree is still an area open for study (Schweik and Semenov, 2003). Weber’s recent book, *The Success of Open Source*, is a significant step in the right direction, arguing, as it does, that OS is a series of experiments in social organization.

Recently scholars have begun to recognize that the FOSS collaborative paradigm is not limited to software (e.g., Bollier, 1999; Schweik and Grove, 2000; Stallman, 2001; CreativeCommons.org, 2004; Schweik and Semenov, 2003; Weber, 2004). It has potential, under the OC licensing approach, to be applied in any intellectual domain that requires a team of thinkers to tackle a problem. For example, OC licensing has already been used to encourage collaboration or “new derivatives” in areas such as music and art (Creativecommons.org, 2004). There are several “wiki” sites that are utilizing the GFDL for the construction of encyclopedias and other online books (e.g., [Wikipedia](http://en.wikipedia.org/), [Wikibooks](http://en.wikipedia.org/wiki/Wikibooks)). There is an emerging movement to apply OC licensing to educational and scientific initiatives as well (see [Science Commons](http://creativecommons.org/projects/science/proposal)). And there are emerging open source software projects that exhibit some of the scientific collaboration characteristics we have described. For example, the [R-Project for Statistical Computing](http://www.r-project.org/) includes review processes for publishing software, online publishing of user-generated documents, author attribution, etc. This paper is an attempt to think through some of the issues that such OC licensing-
based collaborations might encounter, in one scientific area: land-use change modeling (Schweik, Evans and Grove, 2003).

However, it should be stressed that the FOSS collaborative paradigm is not a panacea and in recent years there has been significant hype over it as a phenomenon. Most of the existing studies of FOSS projects are on the success stories such as Linux, Apache Web Server, and a few others. Clearly, many FOSS projects fail for a variety of reasons such as the lack of critical mass of developers, lack of infrastructure, poor governance, or infighting among developers. Many of the nearly 90,000 projects on Sourceforge.net and other sites are probably stalled or dead in terms of active production. But on the other hand, the FOSS collaborative paradigm in software has produced a large number of real collaborative experiments, and there are enough real success stories and interest – including serious industry and government interest – to testify to what we believe to be its very great potential.

Like FOSS collaborations, OS/OC-based collaborations will require experimentation, and many will fail. But the possibility is there that this innovation in collective action, applied to important social-ecological questions – and to many other fields of scholarly inquiry – has the potential to produce important breakthroughs in areas where the benefits of enhanced collaborative effort can make a difference. In the context of land-use change modeling we believe that the move toward an OS/OC paradigm needs to be an evolutionary movement, working within the existing research-publishing paradigm, rather than a revolutionary movement. We are not advocating the destruction of the existing process of peer review and publication. Rather we see modifications to this paradigm whereby new ideas can be incorporated into the traditional form of scientific collaboration. Combinations of OS and OC licensing, the new collaborative infrastructure as we described, along with the advancement of e-journal capabilities, provide the possibility for more rapid progress than is possible within the existing structure of scientific research and publication.

ACKNOWLEDGEMENTS

We greatly appreciate the efforts of at least three anonymous reviewers who provided extremely insightful comments on previous versions of this manuscript. Support for this study was provided by the U.S.D.A. Forest Service’s Burlington Laboratory (4454) and Southern and Northern Global Change Programs; the Cooperative State Research Extension, Education Service, U.S. Department of Agriculture, Massachusetts Agriculture Experiment Station, under Project Number MAS00847; the Center for Policy and Administration and the Department of Natural Resources Conservation at the University of Massachusetts, Amherst and the Center for the Study of Institutions, Population and Environmental Change at Indiana University (NSFSBR 9521918). Finally, this manuscript benefited from discussions with participants at a workshop on open-source development of land-use/landcover change models (see <a href="http:www.lulc.org/bcworkshop_2003/os_lulc_workshop_report_2003.pdf">open source land-use/landcover report</a>) sponsored by the NSF Biocomplexity program.
The findings, recommendations, and opinions expressed are those of the authors and do not necessarily reflect the views of the funding agencies.

LITERATURE CITED


Figure 1. The Trajectory of FL/OS Software Projects.
(Adapted from Schweik and Semenov, 2003;
OC projects too, may follow a similar trajectory)
Figure 2: Structure of the UrbanSim model
(adapted and interpreted from Figures 3 and 4 in Waddell, 2002)
This Appendix provides an overview of the general approach and structure of the UrbanSim model that our OS/OC is focusing on (Schweik, Evans and Grove, 2003). We include this Appendix to give interested readers a little more information on the model to help understand where we are headed with an OS/OC land-use change collaboration. For even more information, on UrbanSim, read Waddell (1998, 2000), Waddell and Ulfarsson (forthcoming), and Noth et al. (2003). This summary is based upon reading these articles.

UrbanSim is described as “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 (http://www.urbansim.org). Consequently, UrbanSim models 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 2 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.