Applying Elinor Ostrom's Rule Classification Framework to the Analysis of Open Source Software Commons

Charles M. Schweik and Meelis Kitsing[*]

Abstract: This research investigates the utility of Ostrom and Crawford's rule classification framework (elaborated in *Understanding Institutional Diversity*) in the systematic study of rule systems in a set of relatively complex open source projects and their overarching non-profit foundation. Using this framework, Rule configurations are described for the overall Open Source Geospatial Foundation and for each of seven associated geospatial projects.

Key words: Ostrom, Crawford, classification, rules, software

1. Introduction

It is an honor for me (lead author Schweik) to be invited to contribute an article to this special issue of Transnational Corporations Review on "Ostroms Studies," celebrating Lin Ostrom's recent Nobel prize in Economics. Like many of the other people who have contributed to this volume, I am a former student of Lin and Vincent's, and one who has studied closely with Lin both in the International Forestry Resources and Institutions research program (IFRI, 2009) initiated at the Workshop in Political Theory and Policy Analysis at Indiana University back in the early 1990s, and in research related to forest change at the Center for the Study of Institutions, Population and Environmental Change (CIPEC, 2009) in the mid-to-late 1990s. Through these efforts, I had the great pleasure of working and being mentored by Lin in the study of environmental commons situations, and in particular, the institutional analysis of natural resource commons.

Prior to this time at Indiana University, I was a computer programmer. In fact, the reason I first began working with Lin was because she hired me as a database developer for the IFRI research program. After I left Indiana University with my PhD in hand, I began to hear the term "open source software." It took me several years to fully make this connection, but around the turn of the century, thanks to my schooling with Lin and Vincent, I realized that open source software was a form of commons; one that is digital and is managed over the Internet. I also concluded then that much of the theory on commons governance from natural resource settings might be informative in understanding how these "new commons" operate. Since

[*] Meelis Kitsing works at the Center for Public Policy and Administration at the University of Massachusetts at Amherst, USA. Charles M. Schweik works at the Department of the Natural Resources Conservation, Center for Public Policy and Administration and National Center for Digital Government at the University of Massachusetts, Amherst. He can be reached at cschweik@pubpol.umass.edu.

then, through support from the United States National Science Foundation, I established a research program to study on open source programming projects – what I call “open source commons.” I have been involved in that effort ever since.

2. An open source primer

For readers who may be unfamiliar with the concept, “open source” refers to computer software code – the logic of the computer programs that make the computer do what it does – that is made available, open access and readable. The innovation of open source can be traced back to the mid-1980s to Richard Stallman, a programmer at the Massachusetts Institute of Technology. Stallman, emphasized that the digital nature of software and the low cost for sharing (particularly across the Internet), meant that code should be treated more as a public good than a private one. Moreover, users of software should also be, by default, given the freedoms to use, read and modify this software and to distribute the software as they deem necessary (Stallman, 1999, 2002). This came to be known as "Free\Libre" (FL) software (Ghosh et al., 2002).

However, what really was Stallman’s brilliant innovation was his use of copyright law to ensure that the software he was working on, called the GNU operating system, granted its users the freedom to:

- run the software;
- read the software source code and modify it;
- redistribute the original version of the software; and
- redistribute modified versions of the program (Stallman, 1999).

Access to the source code is required in order for all but the first right to be realized. These freedoms, when applied through a software copyright license, often mandate that future versions of the software carry the same attributes. Based on these ideas, Stallman created a software license that included these principles referred to as the General Public License or GPL. There are a number of other licenses that have slightly different terms than the GPL, but are considered “GPL-compatible” (Free Software Foundation, 2009b). There are others that are less compatible to the freedoms described above, and the differences often have to do with added restrictions provided in OS license variants that may limit the freedom of users in what they can do with the software (Perens, 1999). The term open source is used for code that falls under this category. But for our purposes, we will use open source to describe all types of software – free/libre or non-GPL compatible open source software.

The freedoms to copy, modify, and distribute readable software source code found in open source provides two potential advantages over the traditional proprietary, full-copyright approach to software development. First, all open source software packages are provided at no monetized cost to the end-user. This creates a powerful incentive for them to be used, particularly in settings where software budgets are small (Hahn, 2002). Second, by providing readable source and permitting new derivations to be created based on this source code, the projects could, in theory, generate a large community of users and developers (Raymond, 2001).

But perhaps most importantly for readers of the Journal of Transnational Corporations Review, the innovation of open source licensing, in conjunction with the growth of Internet-based collaborative tools or platforms (such as Sourceforge.net, an open source software hosting website that currently hosts over 250,000 open source projects) creates new opportunities for transnational collaboration. From this perspective, open source projects are a form of Internet-based commons (Benkler, 2006), but one that differs from the environmental commons that readers here may be familiar with (e.g., Hardin, 1968; Ostrom, 1990). In open source commons, groups of people act collectively to produce a public good, i.e., the software, rather than over-appropriate the resource (e.g., Hardin, 1968).

The key question we have been asking in our research on open source is how do these Internet-based and often transnational collaborations in software operate? What kinds of governance structures and systems of rules are in place and how do they evolve?

The work presented here is a subset of a larger research program that is cumulating in the production of a book, entitled *The Success and Abandonment of Open Source Commons* that we expect to publish sometime in 2011. In this paper, we provide results from one case study of a nonprofit foundation and seven open source projects all in the area of geospatial technologies. Here we investigate the utility of Elinor Ostrom and Susan Crawford's rule classification framework (see Chapter 7 in Ostrom, 2005) to describe the rule systems that are found in this particular network of Internet-based commons.

3. Ostrom's (and Crawford's) rule classification framework

We are making the assumption that many readers of this special issue will already be familiar with the Institutional Analysis and Development framework that has provided a theoretically-based scaffolding that has helped structure many projects undertaken by Lin and her colleagues (Ostrom, 2005:6; Ostrom, Gardner and Walker, 1994). Because of space limitations, we will just summarize briefly IAD here and describe how they apply in open source circumstances (see Schweik, 2005 for more information on IAD applied in an open source context).

In IAD, the "Operational level" is a general name for rules that influence the everyday decisions and actions of project participants. In an open source setting, in part from what we've learned through our ongoing study, these tend to be norms or more formal rules that specify how the further development and support of the software may proceed.

The second institutional level in IAD is referred to as "Collective Choice." This level can be thought of as two general types of rules that "oversee" operational-level rules and structures (Ostrom, Gardner and Walker, 1994). The first type specifies who is eligible to undertake certain operational-level activities. For example, in open source commons, most projects will likely have a norm or rule specifying who has the authority to promote new or revised software code changes to the "next release" library (Fogel and Bar, 2003). In some projects, there may only be one or two people on the team and as a consequence, one or both have this authority. In larger team settings, this authority could be centralized or distributed, depending on the project. The second type of Collective-Choice rules specify who can change operational-level rules and the procedure to make such changes. For example, if a project grows in terms of developer team size, there may be a need to change the Operational-level rule describing how code gets checked in or "committed." Collective-Choice rules would determine how an existing operational procedure would be changed.

The "highest" level in these nested institutions is referred to as "Constitutional-level" rules. One constitutional provision in open source projects is the particular copyright license used (described earlier). But Constitutional-level rules also specify who is allowed to change Collective-Choice rules and the procedures for making such changes. This situation might arise in an open source setting when the recognized leader of a project decides to move on to a new opportunity. Constitutional-level rules could specify who takes over this person's position.

One of the advances Ostrom and her colleague Sue Crawford made in *Understanding Institutional Diversity* (Ostrom, 2005, Chapter 7) was adding further detail or structure to help analysts classify or organize rules found in commons settings at any or all of these levels. In this work, Ostrom and Crawford present seven rule categories: Position, Boundary, Choice, Aggregation, Information, Payoff and Scope. Table 1, Columns 1 and 2 summarize the definitions of each, respectively. In the two sections that follow, we briefly summarize our empirical work investigating how these types of rules fit in the context of an open source "federation" of projects.

Table 1. Ostrom's (2005) seven general rule categories in the OSGeo Foundation's institutional design

Ostrom's Rule Category	**Ostrom's Defintion**	**Examples in OSGeo's Institutional Design**
	Define the positions that participants hold	Board of Directors (BOD); President and CEO; Vice President; Committee

Position **Rules**		Chair; Corporate Officer; Member; Participant
Boundary Rules	Define: (1) who is eligible to take a position (succession rules); (2) the process that determines which participants may enter (entry rules), such as by invitation, through some sort of competition, or compulsory; (3) how an individual can leave a position (exit rules). There may be also rules regarding the relationship between multiple positions, such as a mandate that no one person can hold multiple positions at the same time.	BOD election BOD member leaving Committee Chair Charter v. Other Members
Choice Rules	Specify what participants in positions must, must not or may do in their position and in particular circumstances. Choice rules focus on *actions*.	Bylaws for BOD; Bylaws for Officers Committee rules/policies Incubation process
Aggregation Rules	Determine whether a decision by a single or multiple participants is needed prior to an action at a decision point in a process. Aggregation rules are needed whenever choice rules provide multiple positions partial control over the same sort of actions. Aggregation rules can be symmetric (e.g., unanimity) or nonsymmetric (where a leader can make a decision for a group) and each also must include a non-agreement rule.	Symmetric: Consensus in Committees Nonsymmetric: BOD creates committees
InformationRul	Specify the channels used to communicate information among participants, as well as what kinds of information can be transmitted by what positions. There may also be rules specifying required frequency of interaction, or specifying an official language.	Required Meeting Minutes Required Meeting Notification Annual Meetings Required Financial Statements Required
Payoff Rules	Assign external rewards or sanctions for particular actions or outcomes. For example,	Executive Director and others can be

	some payment for completion of a task.	paid; BOD cannot be paid
Scope Rules	Specify which outcomes may, must, or must not be affected within a situation. Scope rules focus on *outcomes* (compared to choice rules which focus on *actions*).	Organizational Mission Committee Mission

4. The open source geospatial foundation, its associated projects and research methods

There are many types of open source technologies. In some instances, developers from these projects have created nonprofit foundations to, in part, "enable collaboration between a community of individuals and corporate actors" (O'Mahony, 2005:393). Here, we focus our attention on one such nonprofit called the Open Source Geospatial Foundation, which supports open source projects specifically working on geospatial technologies, such as desktop or web-based Geographic Information Systems software. According to the Foundation's website (OSGeo, 2009a), the Open Source Geospatial Foundation is a non-profit organization who supports and promotes the collaborative development of open source geospatial software technologies. The foundation provides financial, legal and organizational support to projects that are formally associated with the organization. It also provides outreach and advocacy services for these software projects.

As of December, 2009, OSGeo lists 21 affiliated projects, some of which are fully "accepted" projects while others are in "incubation," meaning they are working to become fully OSGeo sanctioned projects. In the summer of 2008 we conducted semi-structured elite interviews with representatives from seven of their associated software projects (except in one project which was discontinued or abandoned where we were able to conduct only one interview), and also interviewed the OSGeo Foundation's executive director. Usually, for each specific software project, we interviewed the formal or informal project leader and one other "core developer." We chose this approach because our interview questions were complex and required a good knowledge of institutional history of each project. We used Skype or the telephone to conduct interviews. Interviewees are located in the United States (Alaska, Arizona and Massachusetts), Canada (British Columbia and Ontario), Europe (Poland, Switzerland, Italy, Germany and France) and Australia. In the analysis that follows, we will keep the project names anonymous and we have identified them as A, B, C, D, E, F and G.

5. Using the Ostrom/Crawford Rule classification to articulate OSGeo's system of rules

In the OSGeo context, there are two “scales” of analysis required: rule systems at the Foundation scale, and rule systems at the individual project scale.

5.1. OSGeo Foundation rules

The OSGeo foundation and its projects have a diverse institutional framework consisting of all the rule categories we highlighted above. As the space is limited, we will offer a brief summary of our key findings.

OSGeo is a nonprofit corporation registered in Delaware. The foundation has detailed Constitutional and Collective-Choice level bylaws. Position rules give ultimate power to the “Charter Members,” who elect the Board of Directors (BOD) and vote to admit other Charter Members. In addition to this category, OSGeo acknowledges “Members” and “Participants.” “Members” are people who can participate in the activities of Foundation (e.g., write code, participate in the committees et al.). “Participants” can do all of the same things as Members, but they have not formally self-registered on the foundation’s website (OSGeo, 2009b). According to the choice rules both Members and Participants can participate in projects and committees, but cannot vote for the BOD or being involved in appointing new Charter Members.

The OSGeo constitution outlines choice rules, which grant exclusive right to form committees and nominate their chairs as well as take actions not specified in the bylaws to the BOD. According to a position rule, committee chair must either hold a seat on the BOD or be an officer of the OSGeO. Currently, the foundation has eight committees such as Website, Marketing, Conference and Education among others. Operational-level choice rules stipulate how aspiring open source software projects can become officially affiliated with the OSGeo. These rules highlight concrete steps to be taken in the “incubation” process. Another specific position rule states that official OSGeo projects have a representative on the Incubation Committee, which oversees incubation of phase of aspiring projects. Each incubation project has assigned a mentor who is a member of the Incubation Committee. Hence, software developers familiar with the process or with a prior incubation experience direct each aspiring project in the incubation phase.

Official affiliation with the OSGeo requires each project in the incubation phase to set up a Project Steering Committee (PSC) or Project Management Committee (PMC) that is a legal committee of OSGeo. According to a constitutional boundary rule, the PSC/PMC chairperson must be either a member of the OSGeo BOD or an OSGeo corporate officer. The criteria of being corporate officer is met as well when the BOD appoints the project chair as a designated Vice President of OSGeo (see OSGeo, 2009c). The process seems to put each project under the control of the OSGeo BOD. But in reality the projects maintain most of their freedom and autonomy. For instance, OSGeo projects are encouraged to use

OSGeo provided hosting services (web sites, version control, mailing lists, etc.), but this is not mandated but rather offered as simply a choice.

In sum, OSGeo has created position rules at different IAD levels such as Constitutional, Collective-Choice and Operational to define the positions participants hold (board members, committee chairs, corporate officers and incubation project mentors). Boundary rules set forth processes for entrance into these positions and succession. Choice rules, which clarify of what is expected of what position holders must, may or must not do are found in organizational bylaws and committee rules and policies. There are aggregation rules directing committees and the BOD in decision-making process. In addition, information flow and payoff rules as well as general mission statements for the organization and committees qualify, to some degree, as one form of scope rule.

5.2. Individual OSGeo project rules

Let us now turn to the rules that exist within the seven individual software OSGeo projects that we interviewed. Given our space limitations, we will not detail all identified rules and at specific Operational, Collective-choice or Constitutional levels in IAD. Our intentions here are only to show that these various rules exist and that are can be mapped on the basis of Ostrom's rule categories. Table 2 provides this summary, with each of the seven projects, compared and contrasted, side by side.

Table 2. Ostrom's (2005) rule categories applied to OSGeo project cases

Ostrom's rule category/ project	**Project A**	**Project B**	**Project C**	**Project D**	**Project E**	**Project F**	**Project G**
Position rules	Project Leader Project Steering Committee Member, Core Developer (informal - often overlaps with the Committee Member) Developer	Project Leader Project Steering Committee Member, Core Developer (informal - often overlaps with the Committee Member) Developer	No formal Project Leader, informal lead team of 3 people, Project Steering Committee Member, Committers.	Project Leader Project Steering Committee Member Core Developer (informal - often overlaps with the Committee Member) Developer	Project Leader Project Steering Committee Member, Core Developer (informal - often overlaps with the Committee Member) Developer	Project Leader Project Steering Committee Member Core Developer (informal - often overlaps with the Committee Member) Developer	No formal Project Leader, informal lead team of 4 people, Project Management Committee Member Core Developer (informal - often overlaps with the Committee Member)

Table 2. Ostrom's (2005) rule categories applied to OSGeo project cases

Ostrom's rule category/ project	**Project A**	**Project B**	**Project C**	**Project D**	**Project E**	**Project F**	**Project G**
							Developer
Boundary rules	Formal rules. Community members elect to PSC. No term limits	Formal rules.	Formal rules copied from another project.	Formal rules copied from another project. almost never consulted.	Formal rules.	Formal rules exist but primarily depend on social norms.	Formal rules, but not necessarily followed.
Choice rules	Some formalized. Program Steering Committee makes some major rules, Primarily social norms, open exchange in the list, mutual expectations	Some formalized available in the wiki. Primarily social norms.	Some formalized available in the wiki. Primarily social norms.	Social norms	Social norms	Social norms	Formalized rules written down. Program Management acts if necessary. Social norms important
Aggregation rules	Informal-Symmetric: Consensus in Program Steering Committee and discussion in the including developers who are not in the PSC Formal Voting: Incurs rarely – even though formal rules stipulate Only PSC members can	Steering Committee – almost all developers are on the committee. Voting. If veto vote is used, discussion follows.	Informal-Symmetric: Consensus in Program Steering Committee Formal Voting: Incurs rarely – even though formal rules stipulate Only PSC members can vote.	Steering Committee makes decision by consensus or voting. All developers can vote as well but their vote does not count.	Informal-Symmetric: Consensus in Program Steering Committee Formal Voting: Incurs rarely – even though formal rules stipulate Only PSC members can vote.	Informal-Symmetric: Consensus in Program Steering Committee and discussion but often back channels used before the decision is reached. Voting as a last resort.	Program Management Committee votes

Table 2. Ostrom's (2005) rule categories applied to OSGeo project cases

Ostrom's rule category/ project	Project A	Project B	Project C	Project D	Project E	Project F	Project G
	vote.						
Information rules	Social norm – open exchange of information. Unwritten rule that email list is the main communication tool	Limited formal rules. Most decisions are made in IRC and mailing list is used as well.	Social norms, Talking over email and weekly IRC meeting.	Social norms, project leaders available on IRC almost all the time	Social norms	Social Norms, all communication is based on writing	Social norms Weekly IRC meetings. Otherwise no clear rules.
Payoff rules	No rules	No rules	No rules	No rules	No rules	No rules	No rules
Scope rules	Design rules	Design rules	Design rules	Design rules	Design rules	Design rules	Design rules

Beginning with Position Rules (positions people hold), then there is little variation; all projects have a Project Steering Committee or Project Management Committee (these labels imply the same thing) and an informal group of core developers that often overlaps with the formal management committee. However, there are no formal leaders in two projects (C and G). Project (C) was abandoned in the summer of 2008.

All seven projects have formal Boundary Rules, which stipulate who are eligible and how people enter or leave positions. However, often projects rely on social norms in their governance instead of these formal rules. Similarly, the mapping of Choice Rules, which describe actions people can take in various positions, indicate that project participants follow primarily social norms. Our study of Aggregation Rules concerning how key decisions are made show that all projects have steering or management committees. Formally, these committees use voting as its decision-making method and only committee member votes count. However, most decisions are made through consensus and issues are settled through Internet-based discussion. Most interestingly, in one case, all developers actually vote but only the votes of designated committee members count. Thus, key decisions about project direction are primarily managed via social norms rather than more formal voting structures. According to one project leader, large number of formal rules can de-motivate programmers to take part in the project. In other words, managing the project on the basis of formal rules can make it difficult to keep the current developers and attain new developers, reflecting the idea of "formal rules as friction" (Schweik and English, 2007).

Following the same insight, Information Rules concerning what and how information is communicated and by whom and how often are based on social norms instead of strict formal rules stating which channels will be used. Projects use both email and Internet Relay Chat (IRC). Some projects have weekly IRC meetings. The real difference in project governance is not made by having a norm concerning the communication but rather by the amount of effort contributed to the effective use of communication channels and making vital information available. For example, projects C and D offer principally the same functionality and could be considered competitors. However, project D exploited more effective use of IRC by making one of the core developers available on almost all the time. One key developer is noted for quickly answering questions posed by other developers.

Payoff Rules (external rewards or sanctions) function in an indirect manner and are relatively complex. Most importantly, there is no connection between rewards and project management, i.e. the project leaders cannot decide who gets paid and does not reward completion of particular work. Thus, there are no direct payoff rules. All that project participants can expect to receive stems from social norms and is of non-material nature. For example, successful completion of the task may lead to the reputational gains for a developer. The reward can take a form of a nice email praising one's work (a reputational or signaling related-reward). According to our interviews when a developer's actions have caused problems, an inflow of angry emails serve as a penalty (a negative signal) and should create incentives for them to be more careful in the future. This is well understood in the open source literature and confirms this behavior in the OSGeo context.

All of this does not mean that monetary rewards do not exist and do not matter. Non-material gains such as good reputation may lead to opportunities to get involved in new projects and salaried job or higher salary in the long run. Our interviews show that in five out of seven projects almost all core group members received compensation for their contributions. In these projects, primarily firms but also public sector agencies and/or NGOs reward programmers directly for their work on projects. Only Projects A and E are primarily based on volunteer contributions but in this case some volunteer contributors we interviewed revealed also a very concrete motivation: they used the program in their daily work, and hence, benefited directly from the improvement of software. In the five cases involving direct monetary rewards, developers may have a detailed contract with an employer, which asks them to dedicate a particular percentage of their time to the project. For example, Project D's leader is required to dedicate 20 percent of his time to that project. It was pointed out many times in our interviews that monetary rewards are not the sole motivator and that the compensation is not directly linked to performance; no respondents knew of any penalties or rewards for particular outcomes that developers might receive. The constrained role of monetary compensation is understood further by the fact that many of these cases started as volunteer projects and the success brought the opportunities for the paid work. Our interviews also demonstrated that volunteers who join the existing projects and demonstrate their abilities well often end up as paid contributors.

Scope Rules found in Table 1 are ones that specify which outcomes may, must, or must not be affected within a situation. Scope rules focus on outcomes in comparison to choice rules which focus on actions. Our interviewees pointed out that outcomes depend on the technical design of the project. There were no formal rules related to technical design outcomes. Scope Rules are articulated primarily through informal norms related to software functionality outcomes and are written in standard communication channels.

In short, Table 2 provides a summary of the formal rules and informal norms across all these seven cases. Although all these cases are part of OSGeo foundation framework, variation does exist within most rule categories. As far as Position Rules are concerned the existence or lack thereof of a designated leader appears to make a difference. The two projects without a designated lead face more challenges (one is now abandoned) compared to the others all with established leadership positions. In some instances existing formal Boundary Rules are not consulted or they are overridden by informal rules. There is variance between formal rules and social norms across projects in the Choice, Aggregation and Information rule categories. There were no projects with established Payoff Rules. Scope rules exist and exhibit little variability.

6. Conclusion

Our goal in this short paper was to demonstrate the utility of Ostrom and Crawford’s rule classification system for analyzing the institutional designs of open source projects. This analysis provides convincing evidence that this classification system can be used in these contexts. As the world continues to move toward Internet-based commons to support transnational collective action and collaboration, a standardized classification system to aid in systematically articulating and analyzing institutional designs will become increasingly important. Ostrom and Crawford’s system provide a useful, and we think important, step forward. The challenge is how one can successfully interview and investigate more specifics about established rule systems in a thorough yet efficient method. Readers interested in more information about this particular case and other related research we have done related to this project are encouraged to look for our upcoming book entitled *Success and Abandonment in Open Source Commons* that we hope will be published sometime in 2011.

References

Bar, M., and K. Fogel. 2003. *Open Source Development with CVS.* Paraglyph Press.

Benkler, Yochai. 2006. *The wealth of networks: How social production transforms markets and freedom.* Yale University Press.

Center for the Study of Institutions, Population and Environmental Change. [cited December 9 2009]. http://www.cipec.org/.

Garrett, H. 1968. The tragedy of the commons. *Science* 162, (1968): 1243-8.

Ghosh, R. A., R. Glott, B. Krieger, and B. Robles. 2002. Floss Final Report–Part 4: Survey of developers. *University of Maastricht, the Netherlands www.Infonomics.nl/FLOSS/report*.

Hahn, R. W. 2002. *Government policy toward open source software.* Washington, D.C.: Brookings Institution Press.

International Forestry Resources and Institutions Program. [cited December 9 2009]. http://www.sitemaker.umich.edu/ifri/home.

O'Mahony, S. 2005. Non-profit foundations and their role, in community-firm software collaboration. In *Perspectives on free and open source software*, eds. Joseph Feller, Brian Fitzgerald, Scott A. Hissam and Karim R. Lakhani. Cambridge, Mass.: MIT Press.

Open Source Geospatial Foundation. About the Open Source Geospatial Foundation. [cited March 10 2009]. http://www.osgeo.org/content/foundation/about.html.

———. Membership Rules. [cited October 8 2009]. http://www.osgeo.org/Membership.

———. Open Source Geospatial Foundation Officers and Board of Directors. [cited August 7 2009]. http://www.osgeo.org/content/foundation/board_and_officers.html.

Ostrom, E. 2005. *Understanding institutional diversity.* Princeton University Press.

———. 1991. *Governing the commons: The evolution of institutions for collective action.* Cambridge University Press.

Ostrom, E., R. Gardner, and J. Walker. 1994. *Rules, games, and common-pool resources.* University of Michigan Press.

Perens, B. 1999. The Open Source Definition. In *Open Sources: Voices from the Open Source Revolution,* ed. Chris DiBona, Sam Ockman, and Mark Stone, 53-70. Sebastopol, CA: O'Reilly & Associates, Inc.

Raymond, E. S. 2001. *The Cathedral and the Bazaar: Musings on Linux and open source by an accidental revolutionary.* O'Reilly & Associates, Inc. Sebastopol, CA, USA.

Schweik, C., and R. English. 2007. Conceptualizing the institutional designs of free/libre and open source software projects. *First Monday* 12, (2).

Schweik, C. M. An institutional analysis approach to studying libre software 'Commons'. *Upgrade: The European Journal for the Informatics Professional* Vol. VI, no. 3 (June), www.upgrade-cepis.org/issues/2005/3/up6-3Schweik.pdf.

Stallman, R. 1999. The GNU operating system and the free software movement. In *Open Sources: Voices from the Open Source Revolution*, ed. Chris DiBona, Sam Ockman, and Mark Stone, 53-70. Sebastopol, CA: O'Reilly & Associates, Inc.

Stallman, R. M. 2002. *Free software, free society: Selected essays of Richard M. Stallman*. Boston, MA: Free Software Foundation.

About the Author: Charles M. Schweik (Charlie) is an Associate Professor with a joint appointment to the Department of Natural Resources Conservation (www.umass.edu/nrc) and the Center for Public Policy and Administration (www.masspolicy.org) at the University of Massachusetts, Amherst. He is also the Associate Director of the National Center for Digital Government (www.ncdg.org), and an affiliated researcher with the Science, Technology, and Society Initiative at the University of Massachusetts, Amherst. His research focuses on environmental management and policy, public sector information technology, and the intersection of those domains.

Contact Information

Department of Natural Resources Conservation, University of Massachusetts
Room 217, 160 Holdsworth Way, Amherst, MA 01003-9285 Tel: +1 413 545 1824 Fax: +1 413 545 4358
Email: cschweik@pubpol.umass.edu.