J Geogr Syst (2009) 11:191–207
DOI 10.1007/s10109-009-0086-8

ORIGINAL ARTICLE

Show me the code: spatial analysis and open source

Sergio J. Rey

Received: 25 February 2009 / Accepted: 2 March 2009 / Published online: 10 April 2009

Abstract This paper considers the intersection of academic spatial analysis with the open source revolution. Its basic premise is that the potential for cross-fertilization between the two is rich, yet some misperceptions about these two communities pose challenges to realizing these opportunities. The paper provides a primer on the open source movement for academicians with an eye towards correcting these misperceptions. It identifies a number of ways in which increased adoption of open source practices in spatial analysis can enhance the development of the next generation of tools and the wider practice of scientific research and education.

Keywords Open source · Spatial analysis

JEL Classification C21 · C88

1 Introduction

This paper considers the intersection of two communities, the first consists of the field of academic spatial analysis (Okabe 2006; Goodchild et al. 2000), while the second is the wider world of open source software (Himanen 2001; Feller et al. 2005b). The primary concern is the engagement of the former with the latter, and the paper's goals are twofold. First, it examines the parallel resurgence in spatial analysis and the rise of the open source movement. In doing so it identifies the key opportunities and challenges that this juncture presents to the academic spatial analysis community. These issues transcend the development of software tools for spatial analysis and have important implications for the future growth of the discipline.

S. J. Rey (✉)
School of Geographical Sciences, Arizona State University, Tempe, AZ, USA
e-mail: sjsrey@gmail.com

A key theme of the paper is that each community has a well developed sense of its traditions and conventions yet the interaction between these worlds has not been as strong as perhaps it could be. Although analogies are often drawn between the logics of the two communities, the analogies are sometimes based on misunderstandings about key aspects of the other community. These misperceptions lead to a number of important challenges that require addressing before the powerful, but latent, synergies between the two communities can be realized.

Therefore, the second goal of the paper is to provide a primer on the culture and operation of open source communities. It presents an overview of the way open source communities emerge, evolve and function. It also attempts to provide the same for open source communities. That is, it provides insights as to the functioning of the community of scholars contributing to advances in the science of spatial analysis. In doing so I will highlight the specific synergies that may be tapped into by the intersection of the two communities.

The paper draws on the author's experience as a member of the community of academic spatial analysis researchers as well as a participant in the world of open source software development. With regard to the former, insights from experiences in the scientific projects Space Time Analysis of Regional Systems (STARS; Rey and Janikas 2006) and A Python Library for Spatial Analytical Methods (PySAL; Rey and Anselin 2007) are discussed.[1] The paper also draws from the author's participation in the Linux and Python communities. By revisiting the relationships between these two communities, the paper identifies a number of ways in which the academic spatial analysis community can reinvent itself to enhance its currency with wider technological and community dynamics fueled by the open source movement.[2]

2 A tale of two communities

Open source has been called the "twentieth century's only true innovative concept in business, representing all that is truly new in the new economy" (Sandred 2001, p. xlii). This is largely because open source collaboration presents a very different model for firms with regard to innovation and product development. More broadly, open source is seen as a revolutionary collection of tools and processes through which individuals create, share, and apply new software and knowledge (Feller et al. 2005a). Many analogies have been drawn between the logics of the open source development model and the way scientific communities function (DiBona et al. 1999). Notions of peer review, building on the shoulders of giants, collaboration, openness, reputation, and standard norms of practice are found in both worlds, yet these analogies gloss over some important distinctions (Kelty 2005). In other words, while open source and science are similar, they are not

[1] STARS is a package supporting exploratory space time analysis of areal unit data which was released in early 2006. PySAL is a collaborative effort between the GeoDA (Anselin et al. 2006) and STARS teams to develop a common library of spatial analysis methods. It is planned for release in March 2009.

[2] While many of the issues raised below have relevance beyond spatial analysis to all of scientific research in academia, I focus on my home discipline.

the same thing, and ignoring the differences may hinder opportunities for cross-fertilization between the two. In what follows I outline the key features of these communities and these important differences.

2.1 Open source and free software

Despite their growing popularity, the concepts of open source software and free software are often assumed to be one and the same, but this misperception confuses some rather important distinctions between the two. According to the Free Software Foundation (FSF) (Free Software Foundation 2008), software is considered free if the users of the program have the freedom to:

- Run the program, for any purpose.
- Modify the program to suit their needs.
- Redistribute copies, either gratis or for a fee.
- Distribute modified versions of the program, so that the community can benefit from their improvements.

A tenet of the free software movement is that because source code is fundamental to the development of the field of computer science, having freely available source code is a necessity for the innovation and progress of the field.

Free software is distinct from freeware or shareware. Both of the latter circulate on the internet and are free for anyone to distribute. Freeware is gratis, however, shareware is released on a trail basis with the expectation that users will pay for the program if it suits their needs. What distinguishes free software from freeware/shareware is that the source code for the latter programs is typically not made available, and therefore users cannot modify the program.

The original coining of the term open source was an attempt to differentiate free software from being confused with freeware/shareware as well as to avoid the perception that *free* implied that the code was of inferior quality and not fit for corporate use. Indeed the Open Source Initiative (Raymond 1988) was created as a marketing program for free software. The free and open source movements now differ primarily on philosophical grounds, with the latter tending to emphasise the practical benefits of open source licenses and the former stressing the moral issues involved. Nevertheless the term “open source” is commonly used to refer to either free software or open source by practitioners and scholars of both movements. Yet it is important to note that, from a set theoretic perspective, free software can be viewed as a subset of open source software in that free software is always open source, yet open source software does not necessarily qualify as free software since it may not guarantee the core freedoms specified by the FSF.

Those freedoms are protected under the general concept of “copyleft” which is a play on copyright. More specifically, copyleft attaches copyright to free software together with additional distribution terms that bind the code and the core freedoms together through the GNU General Public License (GPL) (Free Software Foundation 2007b).[3] The GPL is but one of many types of open source licenses and

[3] GNU is a recursive acronym for “GNU Not Unix”.

the issue of license proliferation in the open source movement has been a growing concern (Rosen 2004).

2.2 Open source development logics

While the legal framework provides protections for the freedoms associated with free and open source software, the realization of these freedoms is embodied in the development of programs and software code. The organization of software teams in open source projects is radically different from traditional development models. The canonical comparison of these two models is by Raymond (1999). The traditional model is likened to the manner in which cathedrals were built, consisting of tightly-knit group of developers work closely together on a program, isolated from the external world. The program is not released to the wider world until it has reached a polished stage of maturity.

Contrasting this traditional model is what Raymond saw happening in the world of Linux kernel development, where the project leader Linus Torvald's strategy was releasing code early and often, relying on a large amount of delegation and being open to external input. The larger community of far-flung bands of largely volunteer kernel programmers developing through a process of network collaboration seemed more akin to a market bazaar.

The key difference between the cathedral and bazaar models lies not in whether the code is made available, but rather in the manner in which the code development is organized. The cathedral model is more centralized and requires an a priori approach to design, while the bazaar approach is evolutionary and distributed. That the latter model worked at all seemed to Raymond a small miracle. That it has scaled to projects involving hundreds of developers and which are responsible for much of the software powering today's internet indicates that the process works remarkably well. The success of these projects is not simply because that the code is openly available, but also because of the creation, nurturing and growth of communities of shared interests.

It is important to realize that the cathedral versus bazaar model is a broad contrast between the commercial and open source approaches to software development, and like any abstraction it omits details and variation. For example, as we will see shortly, there are cases of proprietary software houses adopting practices which have their origins in open source projects. At the same time, some have argued that the characterization of open source projects as being self-organizing is inaccurate (Connell 2000), as many of the prominent projects use strong central control, or the so called Benevolent Dictator For Life (BDFL) model to manage large groups of volunteers. With these qualifications in mind, Table 1 summarizes some of the key distinguishing characteristics of open source and proprietary software.

The largely volunteer nature of the open source model raises the obvious question about the motivations of individual developers to join a project. Surveys of open source developers reveal the following reasons given for joining a project: (1) acquisition of new skills; (2) sharing of their knowledge and skills with other developers; (3) participate in new forms of cooperation associated with open source projects; and (4) develop improved software products (Ghosh et al. 2002).

Table 1 Comparison of proprietary and open source software

	Advantages	Disadvantages
Proprietary	Vendor warranty of quality	Purchase and licensing fees
	Customer support	Vendor lock-in
		Access to source code limited
		Limited customization permitted
Open source	Zero purchase and licensing costs	Technical requirements
	Full access to source code	Project forking
	Open standards	Rapid update cycles
	Community support	Documentation can be limited
	Unrestricted customization permitted	

Some of these motivations reflect the characterization of the open source community as a "gift economy" (Cheal 1988). That is, one's standing in the community is not a function of what one owns and controls but rather how much one shares or gives to the community. The community also functions as a technical meritocracy whereby one gains prominence in the community through serving as a developer, project leader, speaker, or writer (Pavlicek 2000). In this community developer reputation serves as the currency (Raymond 1999).

The user community becomes a critical part of the open source development process. Through continuous feedback channels where users can identify bugs, requests features, and provide help to other users, they come to gain an increased sense of responsibility for the software. As they see their suggestions and feedback reflected in new versions of the program, that sense of responsibility can grow into a sense of community ownership. Open source communities are a prime example of what von Hippel (2004, p. 93) labels as *innovation communities*. These communities become sources of user-led innovation, whereby new changes in process and products are increasingly developed by users aided by improvements in computing and communication technology. Agile companies and industries have been able to support and tap into this type of innovation by providing user communities with toolkits for developing new products (von Hippel 2004, p. 14).[4]

While the review thus far has stressed the substantial strengths of open source, it is important to consider the criticisms that have been made of this development model. These criticism surrounds the "developer-centric" nature of open source projects, which can foster technological elitism in the sense that only those individuals with adequate programming skills can participate in the development. This can also result in program interfaces being designed by and for the engineer and developers, rather than with the end-user in mind. Similarly the lack, or poor quality, of documentation in some open source projects presents problems for both non-technical users of the code and may keep potential new developers from joining the project.

[4] One could interpret ESRI's adoption of Python as its scripting language as a similar strategic move to tap into user-led innovation.

In addition, the evolution of software in an open source project can pose problems for adopters. Following the mantra of “release early and release often” an open source project can change very rapidly which can make it challenging to design course curricula around a moving target. This is also a concern for a long term research project which may want to adopt a particular open source package for its computational tasks. New versions of a package can introduce problems with backwards compatibility and force the research project to rely on older versions of the package. A less common problem is when an open source project undergoes a so called “fork” in which developers take the original source code of a project and start an independent project. Although this is broadly frowned upon in the open source community, it can cause problems for larger efforts that have relied on the project that has been forked.

Several of these criticisms have important implications for the adoption of open source in academic spatial analysis, which I return to later.

2.3 Academic spatial analysis

Advanced spatial analytical methods and their implementation in software has been an active area of research for several decades now, predating the rise of the open source movement. Indeed, the field has reached a level of maturity where these tools are now being widely adopted throughout other disciplines. Several major infrastructure projects funded by federal governments, such as the Center for Spatially Integrated Social Sciences (CSISS) (Goodchild et al. 2000) in the US and the Center for Spatial Information Science (CSIS) (Okabe 2006) in Japan, have been developed to support and enhance this dissemination.

Recent surveys of the state of open source projects in GIS (Ramsey 2007; Steiniger and Bocher 2008) reveal that every level of the spatial data infrastructure stack is now covered by open source projects.[5] A related effort to build a complete index of Open Source/Free GIS related software projects lists some 247 packages at the time of writing (Lewis 2007). Considering that the open source movement is only 10 years old its footprint on the world of GIS is impressive. At the same time, a closer examination of the projects suggests that the contributions have been most heavily concentrated on spatial data and traditional GIS functionality, while open source projects in the areas of advanced spatial analysis, statistics, spatial econometrics and spatial modeling tend to be much less prevalent. These areas sit at the top of the spatial analysis research stack.

There are several possible reasons why spatial analysis software is underrepresented on these lists of open source GIS packages. Research on spatial analytical methods is rapidly evolving and researchers working on this frontier understandably concentrate their energies on the theoretical task of new methods development and publishing these contributions. Implementation of the new methods in a software package that is then maintained and supported would divert that energy. There are

[5] The spatial data infrastructure stack consists of the data gathering, management and processing functions which are required prior to carrying out spatial analysis and modeling.

also substantial challenges to creating a package inside an academic environment. Securing funding for tools development is difficult and is often for short duration.

A second reason stems from the doubts held by some developers of spatial analysis software about the value of open source code and the underlying development model. Levine (2001) argued that once the code for a complex spatial analysis program was no longer the province of a single producer there would be the possibility of security breaches and quality control problems. To be fair, Levine's views have evolved over time as his interaction with the open source community has increased, yet his view in 2001 was emblematic of a larger gulf in understanding that existed between the world of open source and academic spatial analysis. While the gulf has narrowed somewhat, there is still a ways to go.

CSISS provides a portal to spatial analysis tools (Center for Spatially Integrated Social Science 2008). A sample of these tools is reported in Table 2. The diversity in the areas covered by these research efforts is mirrored in an array of software distribution models under which the different packages have been released. This spans the gamut from public domain packages, to commercially available closed-source binaries as well as freely available binaries to code released under the GPL.

While the table is not intended to be an exhaustive census of open source projects in spatial analysis, it is representative. In this regard the prominence of software packages based on the R project (Ihaka and Gentleman 1996) is impressive, as gstat, Splancs, Spdep and Spatstat are contributed packages for spatial analysis available on the R repository. Moreover, these four are only a subset of the packages comprising the R spatial project (Bivand 2008) which is arguably the leading effort in open source spatial analysis.

Table 2 Selected spatial analysis software

Program	Domain	License	Cost	Code
ClustrSeer	Epidemiology	Proprietary	$	No
CrimeStat	Criminology	Educational	Free	No
Fragstats	Ecology	Public domain	Free	Yes
GeoVista Studio	Geocomputation	LGPL	Free	Yes
Gslib	Geostatistics	Academic	$	Yes
gstat	Geostatistics	GPL	Free	Yes
Spatial Econometrics Toolbox	Econometrics	Public domain	Free	Yes
Spatial Statistics Toolbox	Spatial statistics	Public domain	Free	Yes
Splancs	Point patterns	GPL	Free	Yes
Spdep	Econometrics	GPL	Free	Yes
S+ Spatial Stats	Spatial statistics	Proprietary	$	Yes
SAS-Geostats	Spatial statistics	Proprietary	$	Yes
Spatstat	Spatial statistics	GPL	Free	Yes
STARS	ESDA	GPL	Free	Yes
variowin	Geostatistics	Academic	Free	No
winBugs	Spatial statistics	Academic	Free	No

Source: Center for Spatially Integrated Social Science (2008)

In thinking about the role of open source in spatial analysis research it is critical to remember that the development model and license chosen for a piece of software matter. To see this, consider that spatial analysis software can satisfy a number of different functions: (1) application function—the package is used in support of an empirical study; (2) innovation function—the code serves as a framework to develop and extend a body of spatial analysis methods; (3) pedagogical function—the program can be examined by researchers and students interested in developing a deeper understanding of the analytical methods.

It is the combination of code availability and the license of the code which influences the extent to which a package addresses each of these functions. It is clear that all the tools listed in the table are available to the end user and thus could all serve the application function. However, for packages that do not provide source code, the researcher is constrained to the set of available analytical functions implemented. By contrast in the case of an open source package, if a certain research project required an extension to the analytical functionality, a capable user could in fact enhance the package to add the new functionality. In this case the research questions drive the application of spatial analysis software rather than the reverse.

Having access to the code can serve important pedagogical goals and provide transparency to research efforts. Source code allows a student or researcher to peek under the hood and examine the precise implementation of an spatial analytical method. Another strength of open source software is that errors in algorithms can be directly identified by users instead of having to indirectly figure out why incorrect output is being generated.[6] Further modification and enhancement of the methods and code, and their release to the wider scientific community may, however, be restricted or limited in cases where code is available but under a commercial license, while under a free license those activities are fully encouraged. At first glance, public domain software may seem to provide a similar function. However, public domain software is software that has explicitly not been put under copyright. Because of this, there is nothing to prevent a user from taking that code and using it to form the core of a closed-source proprietary package *without any explicit attribution to the original authors*. The loss of attribution is anathema to the critical role that reputation plays in science and innovation.

3 Opportunities

The previous section reveals the impact that the open source movement is having on academic spatial analysis. Yet, the intersection of academic spatial analysis and open source could be deeper, and in this section I highlight several areas where opportunities exist for further development.

[6] See McCullough (1988, 1999) on the issue of identifying hidden problems in statistical software.

3.1 Freedoms

Some of these opportunities relate to two different notions of freedom inherit in open source software: *free as in beer* and *free as in speech*.

The first freedom plays several important instructive roles stemming from the fact that the monetary costs of the software are nil since the code is free for the downloading. This is particularly attractive in public academic settings where increasingly budgets are tight. This free availability also means that students can download the packages on their own computers and are thus freed up to learn anywhere since they are no longer restricted to using site-licenced software installed on laboratory computers. Not only does it give students the ability to "time-shift" their studies and to leave the walled garden of the laboratory, it also makes the code available to a tremendous potential audience of future students—anyone with an internet connection.

Another important instructive role provided by the "free beer" characteristic is that students are able to inspect the code to gain a deeper understanding of how particular algorithms or statistical methods actually are implemented. In this sense seeing the *code as text* can have very powerful pedagogical benefits. The quote by the prominent computer scientist Alan Perlis captures this benefit:

> You think you know when you can learn, are more sure when you can write, even more when you can teach, but certain when you can program (Perlis 1982).

The second freedom associated with open source software is so called "free speech". This is the more fundamental of the two freedoms in that it relates to the users ability not only to examine, but to modify, enhance and release source code to the wider community. This freedom can play a constructive role in the world of academic spatial analysis. By tapping into the power of user-led innovation it can stimulate the engagement of skilled attention so vital to the flourishing of successful open source communities.

Research at the forefront of GIScience requires a solid foundation of computational skills (Worboys and Duckham 2004). The steep learning curve facing students who pursue spatial analysis as a research interest has been cited as a challenge to attracting future generations of spatial scientists (Fotheringham 1993; Rey 2001). The instructive and constructive functions played by open source software could be immensely helpful in addressing these entry costs.

3.2 A new kind of science

Increasingly, funding agencies are requiring proposals have interdisciplinarity at their core (National Science Foundation 2006). This is driven in part by the recognition that the types of problems facing the world today are not going to be addressed from within the intellectual silos of individual disciplines (Sachs 2008). Coupled with this is a growing emphasis on cyberinfrastructure and its ability to support distributed collaboration and sharing of tools, instrumentation and data between scientists from different institutions and disciplines (Atkins et al. 2003).

In this new world, the development of so called middleware plays a central role. Middleware focused on comprehensive data repositories, such as the digital library initiative has been credited with stimulating the rise of vast information repositories and services on the Internet (Atkins et al. 2003, p. 6), while other types of middleware, such as the Network Workbench (Börner 1997), serve as scientific glue and integrate tools from different scientific domains into a flexible and unified framework.

Open source can play a vital role in this new research era. Relying on open standards and programing frameworks facilitates the integration of specialized application programs into scientific middleware. From a substantive perspective, open source code as a way to implement integrated models provides a transparency mechanism that can facilitate communication between scholars from different domains. In essence the code can become the *lingua franca* to expose the current understandings of the logic of the systems under study. Again, code as text is not simply a metaphor for rethinking the way researchers view software for spatial analysis. The idea of reading the code for the purposes of learning goes back to the earliest days of the Linux kernel (Moody 2001, p. 43). I am suggesting the same strategy can be used to facilitate interdisciplinary collaboration.

This new research era is also characterized by the growing complexity of the research questions being posed. Increasingly researchers are relying on numerical simulation for results as closed-form solutions are not available for emerging research questions (Atkins et al. 2003, p. 11). This, in turn, is blurring the roles of software developer and scientist as success as the latter will increasingly require competence in programming. Pushing the envelope of scientific questions will require moving beyond a reliance on closed source software and the ability to create new tools tailored to new questions. Researchers engaging with open source gain the freedom to shape the development of spatial analysis software to suit their needs, rather than having to shape their research agendas to fit the capabilities of closed source programs.

Spatial analysis is as well suited to be a major contributor to the world of interdisciplinary science as it is to have its future evolution shaped by this intersection. Consider the name of the field of Geographic Information Science itself. From one perspective, this could be seen as the set of methods and theories used to analyze data that is geographically referenced. Alternatively, the work by Skupin and Fabrikant (2007) demonstrates there is much to be gained from applying principles and methods of spatialization to the analysis of any type of scientific data. The latter perspective opens up a much broader scope for collaboration between spatial analysis and other fields. The implementation of these novel methods of spatialization in open source toolkits would strongly facilitate this type of collaboration (Lacayo and Skupin 2007).

4 Challenges

Exploiting the rich opportunities that open source affords academic spatial analysis requires that the research community address a number of challenges. These relate to: (1) negotiating commercial and academic networks; (2) academic reward structures; (3) nurturing a network community for spatial analysis;

4.1 Commercial and academic networks

In the evolution of spatial analysis and commercial software, academic research has served as an important source of methodological innovations. Yet, a longstanding lament of the academic spatial analysis community has been that the pace of adoption in the commercial realm has been slower than has the rate of innovation in the research lab (Anselin and Getis 1992). In part the slow uptake of advanced methods by commercial software houses reflects the latter being responsive to commercial markets and that science in general, and spatial analysis in particular, represent only a tiny fraction of those markets (Goodchild 2009).

Some see open source development as a threat to commercial software houses. Here open source packages are viewed as competitors, threatening to eat into their end user markets. However, rather than viewing those user markets as only a source for revenue, those markets can be seen as conversations (Searls and Weinberger 2001) from which ideas for new products, processes and innovations can be gained. Moreover, there are numerous examples of companies engaging with open source projects to their benefit. IBM for example hired several leading Apache developers and in doing so gained assurance that development of the Apache Web server would address the needs of the business community (Pavlicek 2000, p. 39). Even in the archetypical open source project, the Linux kernel, it is estimated that over 70 percent of the development is by contributors who are being paid for their work (Searls 2008, p. 15).

These challenges boil down to the question of what the structure of these networks will look like going forward. Are they unidirectional graphs with brains, ideas and algorithms originating in academic spatial analysis but moving to the destination node of the private sector? Or, will the two communities come together to form bidirectional graphs where this type of migration is mirrored in the infusion of support to academic research projects from the private sector? That funding could lead to further innovation and commercialization while also bringing badly needed financial support to university research operations.

While the relationship between commercial software companies and spatial analysis is important, there are also some challenges posed by internal academic networks that require consideration. As mentioned earlier, merit in open source is based on what you have done and contributed. The community is highly agnostic when it comes to an individual's age, gender, nationality or political affiliation. This results in a very fluid community where capable developers are free to leave and join projects as their interests dictate. This contrasts with relatively lower upward mobility found in academic communities where institutional hierarchies can play a fundamental role in shaping community interaction. For example, the perceived quality of the institution granting an individual's doctoral degree can have an important segmentation effect in academic labor markets and can place an upward bound on the future academic mobility of that individual (Ault et al. 1979, p. 152). Although this filtering mechanism may bring efficiencies to departments doing job searches, it may have a repulsive effect on highly talented scholars resulting in their leaving academia.

This potential brain drain is particularly worrisome in a world where academia will increasingly compete for the brightest minds. Gone are the days when academic research labs could claim exclusive dominion over interesting and challenging problems. Exciting research is increasingly being carried out in the private sector often in companies that offer very attractive work environments and who have deep pockets that have been used to lure top talent away from competitor companies (Battelle 2006). Although I am not aware of any formal study of these patterns, I have seen first hand a number of promising doctoral and master students in spatial analysis decide to leave academia to enter the private sector software industry. Conversations with colleagues at other institutions tell me that I am not alone in this experience.

A final point to keep in mind is that while the potential for cross-fertilization between the open source movement and academic research on spatial analysis is promising, it is by no means inevitable. Indeed two of the leading figures in the origins of open source were initially motivated by disenchantments with universities. Richard Stallman, the founder of the FSF, quit his job at the artificial intelligence lab at MIT due to the increasing commercialization of the university software research (Williams 2002). Torvald's himself started his kernel project in part out of numerous disappointments with operating systems research in academia (Torvalds and Diamond 1999, p. 62).

4.2 Academic reward structures

Within academia there are some features of the reward structure system that are somewhat in conflict with the logic of open source. The first relates to spatial analysis tools being viewed as means to an end in research rather than research itself. In other words, the code is not viewed as text in the sense that the development of software for spatial analysis is generally not given the same credit for professional growth as would journal publications or funded research proposals. The general perception is that the application of these tools to address substantive research problems is where new scholars best invest their time. Given the opportunity costs this perception is understandable, however, it does present a sort of internal constraint on new developers within academia.

Closely related to this is the undervaluation of community infrastructure so vital to the flourishing of any type of open source or scientific community. Individual scholars who support, maintain, and contribute to mailing lists and forums associated with a project do not typically have such efforts translate into recognition by promotion and tenure committees. By the same token documentation for open source software in academia is perceived to be scarce, and as mentioned earlier this is often seen as a weakness of open source. This is because of the discordance between the high value that end users place on quality documentation on the one hand,[7] and on the other the perception by evaluation committees that such documents represent

[7] In addition to its excellent functionality of state of the art spatial analysis, one of the important reasons for the impressive popularity of GeoDa (Anselin et al. 2006) is the free availability of detailed user documentation.

scientific grey literature at best. Simply put, the value of a software manual is swamped by that of a refereed journal article.

As mentioned earlier, open source and science are often seen as heavily dependent upon peer review. While this is true, the articulation of peer review is different in the two worlds. Scientific journals typically rely on a double-blind peer review process in which a referee's identity is not revealed to the author, or vice versa. This system protects the referee and ensures candor in the evaluation process. At the same time, referees often make constructive criticisms of manuscripts that can lead to substantial improvements in the ultimate paper. They do this out of a sense of professional obligation as their contributions are rarely attributed to the referee by name.

In the open source world, the code is the analogue to the manuscript and the wider community serves the refereeing function. Here, however, the interaction between the reviewer and the developer is not anonymous and the filing of bug reports, posts to mailing lists, and discussions at conferences are all very public. As a result the contributions of the reviewer to improving the code are attributed to the individual.

In each of these cases there is a disconnection between an open source academician's contribution and the attribution she fails to receive for that contribution. Given that the private (i.e., to the individual scholar) returns of these contributions are generally much smaller than the realized community benefits, it is not surprising that they are under supplied in academia. As a result, we currently have a situation where a large number of researchers are *using* open source code in support of their research projects, while a much smaller minority of scholars are *doing* open source in ways that transform the research and scholarly process.

4.3 Community building and perceptions

These benefits are under furnished largely because the community has not yet reached threshold numbers. The challenge then becomes how to grow the community of open source practitioners in academic spatial analysis. In addition to the issues related to reward structures, there are several perceptions that would need to be changed for these communities to start to scale.

One of these perceptions relates to the brain drain facing spatial analysis. Potential students perceive that the entry costs to becoming a contributor to the field are steep. In part this perception is sometimes perpetuated by members of the clan who themselves successfully climbed the learning curve and are understandably proud of those achievements. However, there is also somewhat of a masochistic characteristic to code development in the spatial analysis community where program execution speed is seen as paramount and therefore low level compiled languages such as C, C++, and Fortran are viewed as the languages of choice for "real computational scientists".

In this regard, the development speed advantages of scripting languages for rapid prototyping have been recognized for some time now in the field of computer science (Kernighan 1995), yet the adoption of this type of programming has been until recently rather slow in spatial analysis. Changing the mind set to give priority to developer (scientist) time over execution time would do much to lower some of the entry costs. Moreover, these higher level languages, such as Python, Ruby and Perl, allow the developer to work closer to the substantive problem domain and, perhaps more importantly, are simply much more fun.

Fun should not be underrated. Programming can be as much art as science, and the creative process it engages can attract the motivated student to work on open source spatial analysis software. I have seen first-hand how the skills learned in working on an open source project are later used to support other non-software related research projects. Open source development tools, including issue and bug tracking systems and code versioning systems have been applied to coordinate collaborative manuscript preparation, shared bibliographies for specialized areas and general scheduling/coordination of large teams of scholars on research projects. More importantly, the collaborative norms students are exposed to in working on an open source project have time and again created positive spillover effects in building a community of scholars both within individual departments as well as across institutions.

A second constraint on the growth of open source spatial analysis community is a perceived conflict between what could be seen as the rent seeking behavior of a scientist and the aggregate welfare of the wider community. Developing a new statistical method that is then implemented in a closed-source package is one strategy to maintain control over the method and to ensure credit where credit is due. An implicit assumption behind this strategy is that the researcher risks a loss of attribution if the original source code were revealed to the broader scientific community. As we have already seen, open source licenses can provide important protections in this regard, and work by von Hippel (2004) reveals that free revealing of proprietary innovations may lead to increases in the innovator's private profit. A related assumption is that free riders in an open source world reap benefits that equal those going to the contributors of innovations, yet contributors to a public good can and do gain larger private benefits than free riders (von Hippel 2004, p. 91).

The growth of distributed GIS (Peng and Tsou 2003) and web services in spatial analysis (Anselin et al. 2004) raises an important challenge in this regard. On the one hand this "software as a service" (SaaS) model of delivery provides end users with access to advanced spatial analytical methods via a web browser and will play a critical role in the dissemination of these methods across the social and physical sciences. On the other hand, it is the service and not the software that is made available to the end user in the traditional SaaS model, and thus the possibility for the community to examine the underlying implementation and contribute to its advancement are not supported. Indeed, Google has been criticized in the open source community for using open source code for many of its on-line services (i.e., Google Earth, Gmail, Google Calendar) but not making the source code for those implementations available. This so-called ASP-loophole has resulted in the GNU Affero General Public License (Free Software Foundation 2007a) which is designed specifically to ensure that modifications of open source web-based implementations become available to the community.

The release of innovative ideas in the form of open source code can widen the scope of ownership and facilitate the growth of a vibrant community surrounding an open source project. Having good code is a necessary but not sufficient condition for project success. The most successful open source projects are those not only with excellent code bases but thriving communities of users and developers. The same holds for spatial analysis.

5 Conclusion

This paper has examined the intersection of academic spatial analysis and the open source software revolution. The intent has been to provide a better understanding of the internal dynamics of the two communities with an eye towards facilitating their cross-fertilization. While that cross-fertilization offers much potential it is by no means inevitable as a number of pressing challenges stand before us.

This review of open source projects in spatial analysis suggests that inroads are indeed being made. We are at a stage where a growing number of researchers are using open source software in support of their research. The full cross-fertilization between the two communities will come only when the number of producers of open source code also begins to grow along side the consumers of such projects. Only then will the community dynamics and reward structures align themselves to have network effects that can transform scientific practice.

Addressing those challenges is likely to lead to new forms of research that rest on the foundation of a scientific commons for spatial analysis. One in which academic research is able to tap into the broader world of user-inspired innovation in research methods. In the end, the reason for examining the logics of the open source movement lays not in its fascinating social dynamics, but in the promise of new ways to organize science and heighten the pace of knowledge discovery.

Acknowledgments Portions of this research were supported by National Science Foundation Grants BCS-0602581 and BCS-0433132. Previous versions of this paper were presented at the 2007 Association of American Geographers Meetings, the University of Southern California, Arizona State University and San Diego State University, where I received many valuable comments that have improved the arguments. I have also benefitted from the suggestions and comments of the anonymous reviewers and the Editors.

References

Anselin L, Getis A (1992) Spatial statistical analysis and geographic information systems. Ann Reg Sci 26:19–33

Anselin L, Kim YW, Syabri I (2004) Web-based analytical tools for the exploration of spatial data. J Geogr Syst 6(2):197–218

Anselin L, Syabri I, Kho Y (2006) GeoDa: an introduction to spatial data analysis. Geogr Anal 38:5–22

Atkins DE, Droegemeier KK, Feldman SI, Garcia-Molina, H., Klein ML, Messerschmitt DG, Messina P, Ostriker JP, Wright MH (2003) Revolutionizing science and engeineering through cyberinfrastructure. Technical report, National Science Foundation. http://www.nsf.gov/cise/sci/reports/atkins.pdf;l

Ault DE, Rutman GL, Stevenson T (1979) Mobility in the labor market for academic economists. Am Econ Rev 69(2):148–153

Battelle J (2006) The search: how Google and its rivals rewrote the rules of business and transformed our culture. Penguin Group, New York

Bivand R (2008) R spatial projects. http://sal.uiuc.edu/csiss/Rgeo//

Börner K (1997) Network workbench: a ci-marketplace for network scientists. In: Third International conference on e-social science, Ann Arbor, MI

Center for Spatially Integrated Social Science (2008) Select spatial tools. http://www.csiss.org/clearinghouse/select-tools.php

Cheal D (1988) The gift economy. Routledge, London

Connell C (2000) Open source projects manage themselves? Dream on. IBM/Lotus Developers Network Archives. http://www.chu-3.com/pub/manage_themselves.htm

DiBona C, Ockman S, Stone M (1999) Opensources: voices from the open source revolution. O'Reilly, Sebastopol

Feller J, Fitzgerald B, Hissam SA, Lakhami KR (2005a) Introduction. In: Perspectives on free and open source software. MIT Press, Cambridge

Feller J, Fitzgerald B, Hissam SA, Lakhami KR (2005b) Perspectives on free and open source software. MIT Press, Cambridge

Fotheringham AS (1993) On the future of spatial analysis: the role of GIS. Environ Plan A 25:30–34

Free Software Foundation (2007a) GNU Affero general public license. http://www.fsf.org/licensing/licenses/agpl-3.0.html

Free Software Foundation (2007b) GNU general public license. http://www.gnu.org/copyleft/gpl.html

Free Software Foundation (2008) What is free software? http://www.fsf.org/about/what-is-free-software

Ghosh RA, Glott R, Krieger B, Robles G (2002) Free/libre and open source software: survey and study. Technical report, International Institute of Infonomics, University of Maastricht. http://www.infonomics.nl/FLOSS/report/

Goodchild MF (2009) Whose hand on the tiller?: revisiting "spatial spatial statistical analysis and GIS". In: Anselin L, Rey SJ (eds) Perspectives on spatial data analysis. Springer, Berlin

Goodchild MF, Anselin L, Appelbaum RP, Harthorn BH (2000) Toward spatially integrated social science. Int Reg Sci Rev 23:139–159

Himanen P (2001) The Hacker Ethic and the spirit of the information age. Secker and Warburg, London

Ihaka R, Gentleman R (1996) R: A language for data analysis and graphics. J Comput Graph Stat 5(3):299–314

Kelty C (2005) Free science. In: Feller J, Fitzgerald B, Hissam SA, Lakhami KR (eds) Perspectives on free and open source software. MIT Press, Cambridge, pp 415–431

Kernighan BW (1995) Experience with tcl/tk for scientific and engineering visualization. In: Proceedings Tcl Tk Workshop, pp 269–278

Lacayo M, Skupin A (2007) A GIS-based visualization module for self-organizing maps. In: Proceedings of 23rd international cartographic conference, Moscow

Levine N (2001) The crimestat program: characteristics, use, and audience. In: Anselin L, Rey SJ (eds) Spatial data analysis software tools. UC Santa Barbara, Santa Barbara

Lewis B (2007) Open source GIS. http://opensourcegis.org/

McCullough BD (1988) Assessing the reliability of statistical software: part i. Am Stat 52:358–366

McCullough BD (1999) Assessing the reliability of statistical software: part ii. Am Stat 53:149–159

Moody G (2001) Rebel Code: Linux and the open source revolution. Penguin Press, London

National Science Foundation (2006) Human and social dynamics: competition for FY 2007. http://www.nsf.gov/pubs/2006/nsf06604/nsf06604.htm

Okabe A (2006) GIS-based studies in the humanities and social sciences. CRC Press, Boca Raton

Pavlicek RC (2000) Embracing insanity: open souce software development. SAMS, Indianapolis

Peng ZR, Tsou MH (2003) Internet GIS: distributed geographic information services for the internet and wireless networks. Wiley, New York

Perlis AJ (1982) Epigrams in programming. SIGPLAN Notices 17:7–13

Ramsey P (2007) A survey of open source GIS. In: Free and open source software for geospatial. Victorial, Canada. http://www.foss4g2007.org/presentations/view.php?abstract_id=136

Raymond ES (1988) OSI launch announcement. http://www.opensource.org/pressreleases/osi-launch.php

Raymond ES (1999) The cathedral and the bazaar. O'Reilly, Sebastopol

Rey S (2001) Mathematical modeling in human geography. In: Smelser NJ, Baltes PB (eds) International encyclopedia of the social & behavioral sciences, vol 14, Elsevier, New York, pp 9393–9399

Rey SJ, Anselin L (2007) Pysal: A python library of spatial analytical methods. Rev Reg Stud 37:5–27

Rey SJ, Janikas MV (2006) STARS: space-time analysis of regional systems. Geogr Anal 38(1):67–86

Rosen L (2004) Open source licensing: software freedom and intellectual property law. Prentice Hall, New York

Sachs JD (2008) Common Wealth: economics for a crowded planet. Penguin Press, HC

Sandred J (2001) Managing open source projects. Wiley, New York

Searls D (2008) Kernel candy. Linux J 171:15

Searls D, Weinberger D (2001) Markets are conversations. In: Locke C, Levine R, Searls D, Weinberger D (eds) The Cluetrain Manifesto: the end of business as usual. Basic Books, Cambridge, pp 75–114

Skupin A, Fabrikant S (2007) Spatialization. In: Wilson J, Fotheringham S (eds) The handbook of geographical information science. Blackwell, London, pp 61–79

Steiniger S, Bocher E (2008) An overview on current free and open source desktop GIS developments. Int J Geogr Inf Sci (in press)

Torvalds L, Diamond D (1999) Just for fun: the story of an accidental revoultionary. Harper Business, New York

von Hippel E (2004) Democratizing innovation. MIT Press, Cambridge

Williams S (2002) Free as in Freedom: Richard Stallman's crusade for free software. O'Reilly, Sebastopol

Worboys M, Duckham M (2004) GIS, a computing perspective. CRC Press, Boca Raton