Towards a Global Research Infrastructure for Multidisciplinary Study of Free/Open Source Software Development

Les Gasser[1,2] and Walt Scacchi[2]
[1]Graduate School of Library and Information Science,
University of Illinois at Urbana/Champaign,
[2]Institute for Software Research
University of California Irvine,
Irvine, CA 92697-3455 USA
+1-949-824-4130, +1-949-824-1715 (fax)
gasser@uiuc.edu, wscacchi@uci.edu

Abstract. The Free/Open Source Software (F/OSS) research community is growing across and within multiple disciplines. This community faces a new and unusual situation. The traditional difficulties of gathering enough empirical data have been replaced by issues of dealing with enormous amounts of freely available public data from many disparate sources (online discussion forums, source code directories, bug reports, OSS Web portals, etc.). Consequently, these data are being discovered, gathered, analyzed, and used to support multidisciplinary research. However at present, no means exist for assembling these data under common access points and frameworks for comparative, longitudinal, and collaborative research across disciplines. Gathering and maintaining large F/OSS data collections reliably and making them usable present several research challenges. For example, current projects usually rely on direct access to, and mining of raw data from groups that generate it, and both of these methods require unique effort for each new corpus, or even for updating existing corpora. In this paper, we identify several needs and critical factors in F/OSS empirical research across disciplines, and suggest recommendations for design of a global research infrastructure for multi-disciplinary research into F/OSS development.

1. Introduction

A significant group of software researchers is beginning to investigate large software projects empirically, using freely available data from F/OSS projects. A body of recent work point out the need for community-wide data collections and research infrastructure to expand the depth and breadth of empirical F/OSS research, and several initial proposals have been made [6, 16, 17, 24, 28]. Most importantly, these data collections and proposed infrastructure are intended to support an active and growing community F/OSS scholars addressing contemporary issues and theoretical foundations in disciplines that include anthropology, economics, informatics

(computer-supported cooperative work), information systems, library and information science, management of technology and innovation, law, organization science, policy science, and software engineering. Approximately 200 published studies can be found at the MIT Free/Open Source research community Web site (http://opensource.mit.edu/). Furthermore, this research community has researchers based in Asia, Europe, North America, South America, and the South Pacific, thus denoting its international and global membership. Consequently, the research community engaged in empirical studies of F/OSS can be recognized as another member in the growing movement for interdisciplinary software engineering research.

This report attempts to justify and clarify the need for community-wide, sharable research infrastructure and collections of F/OSS data. We review the general case for empirical research on software repositories, articulate some specific current barriers to this empirical research approach, and sketch several community-wide options with the potential to address some of the most critical barriers. First, we review the range of research and research questions that could benefit from a research infrastructure and data collections. Second, we expose critical requirements of such a project. We then suggest a set of components that address these requirements, and put forth several specific recommendations.

2. Objects of Study and Research Questions

As an organizing framework, we identify five main *objects of study*--that is, things whose characteristics researchers are trying to describe and explain--in F/OSS-based empirical software research: *software artifacts and source code*, *software processes*, *development projects and communities*, and *participants' knowledge*. In Table 1 we provide a rough map of some representative characteristics that have been investigated for each of these objects of study, and show some critical factors that researchers have begun linking to these characteristics as explanations. It is important to point out that these objects of study are by no means independent from one another. They should be considered as interdependent elements of F/OSS (e.g., knowledge and processes affect artifacts, communities affect processes, etc.) Also, each of the outcomes shown in Table 1 may play a role as a critical factor in the other categories.

3. Current Research Approaches

We have identified at least four alternative approaches in empirical research on the objects and factors in Table 1 [cf. 50,51]:

- *Very large, population-scale studies* examining common objects selected and extracted from hundreds to tens-of-thousands of F/OSS projects [14, 24, 29, 32, 37] or surveys of comparable numbers of F/OSS developers [21, 25].
- *Large-scale cross-analyses of project and artifact characteristics*, such as code size and code change evolution, development group size, composition and organization, or development processes [7, 18, 23, 39].

- *Medium-scale comparative studies* across multiple kinds of F/OSS projects within different communities or software system types [2, 31, 49, 53].
- *Smaller-scale in-depth case studies* of specific F/OSS practices and processes, for concept/hypothesis development and exposing mechanism details [10, 30, 40, 44, 48, 50, 59].

Objects	Success Measures	Critical Driving Factors
Source Code and Artifacts	Quality, downloads, reliability, usability, durability, fit, structure, growth, diversity, localization	Size, complexity, bugs and features, software architecture (structure, substrates, modularity, versions, infrastructure), information architecture, artifact types, and document genres
Processes	Efficiency, ease of improvement, adaptability, effectiveness, complexity, manageability, predictability	Size, distribution, collaboration, knowledge/information management, artifact structure, configuration, agility, innovativeness
Projects	Type, size, duration, number of participants, number of software versions released	Development platforms, tools supporting development and project coordination, software imported from elsewhere, social networks, roles and role migration paths, leadership and core developers, socio-technical vision
Communities	Ease of creation, sustainability, trust, increased social capital, lower rate of participant turnover	Size, economic setting, organizational architecture, behaviors, incentive structures, institutional forms, motivation, participation, core values, common-pool resources, public goods
Knowledge	Creation, codification, use, need, management	Tools, conventions, norms, social structures, technical content, acquisition, representations, reproduction, application

Table 1: Characteristics of empirical F/OSS studies.

These four alternatives are separated less by fundamental differences in objectives than by technical limitations in existing tools and methods, or by the socio-technical research constraints associated with qualitative ethnographic research methods versus quantitative survey research. For example, qualitative analyses are hard to implement on a large scale, and quantitative methods have to rely on uniform, easily processed data. We believe these distinctions are becoming

increasingly blurred as researchers develop and use more sophisticated analysis and modeling tools [e.g., 30, 36, 45, 47], leading to finer gradations in empirical data needs.

4. Available Empirical Data

Increasingly, F/OSS researchers have access to very large quantities and varieties of data, as most of the activity of F/OSS groups is carried on through persistent electronic media whose contents are open and freely available. The variety of data is manifested in several ways.

First, data vary in *content*, with types such as communications (threaded discussions, chats, digests, Web pages, Wikis/Blogs), documentation (user and developer documentation, HOWTO tutorials, FAQs), development data (source code, bug reports, design documents, attributed file directory structures, CVS check-in logs) [48, 51], and programming languages [7].

Second, data originates from different *types of repository* sources [8, 14, 24, 27, 40]. These include shared file systems, communication systems, version control systems, issue tracking systems, content management systems, multi-project FOSS portals (SourceForge.net, Freshmeat.net, Savannah.org, Advogato.org, Tigris.org, etc.), collaborative development or project management environments, FOSS code indexes or link servers (free-soft.org, LinuxLinks.com), search engines (Google.com/code, krugle.org), and others. Each type and instance of such a data repository may differ in the storage data model (relational, object-oriented, hierarchical, network), application data model (data definition schemas), data formats, data type semantics, and conflicts in data model namespaces (due to synonyms and homonyms), modeled, or derived data dependencies. Consequently, data from FOSS repositories is typically heterogeneous and difficult to integrate beyond semantic hypertext linking [41], rather than homogeneous and comparatively easy to integrate.

Third, data can be found from various spatial and temporal *locations*, such as community Web sites, software repositories and indexes, and individual FOSS project Web sites. Data may also be located within secondary sources appearing in research papers or paper collections (e.g., MIT FOSS research paper repository at http://opensource.mit.edu), where researchers have published some form of their data set within a publication [40, 52, 60].

Fourth, different *types of data extraction tools and interfaces* (query languages, application program interfaces, Open Data Base Connectors, command shells, embedded scripting languages, or object request brokers) are needed to select, extract, categorize, and other activities that mine, gather, and prepare data from one or more sources for further analysis [15, 18, 30, 32, 45, 47].

Last, most FOSS project data is available as *artifacts or byproducts* of development, usage, or maintenance activities in FOSS communities. These artifacts/byproducts are a critical part of the FOSS innovation process [61]. However, very little data is directly available in forms specifically intended for

research use. This artifact/byproduct origin has several implications for the needs expressed above [16, 49, 51].

5. Addressing Issues with Empirical Data

The main objective of a research infrastructure that collects, aggregates, organizes, and offer data analysis services is to address community-wide resource issues in community-specific way [1, 58]. For F/OSS research, the objective is to improve the collective productivity of software research by lowering the access cost and effort for data that will address the critical questions of software development research. In this section we offer some possible approaches to such an infrastructure, by first briefly describing each "component", and then considering its benefits and drawbacks.

5.1 Metadata and Meta-models

One of the broadest approaches to common infrastructure is the use of representation standards [1, 59] as meta-data. Such standards would move some issues of cross-source data normalization forward in the process that produces F/OSS projects' information. The use of metadata permits researchers to identify relevant characteristics of specific data collections. Metadata can serve numerous roles in the organization and access of scientific data and documents, including roles in location, identification, security/access control, preservation, and collocation [54]. Standardization of metadata and addition of metadata to F/OSS information repositories, especially at the point of creation, would let the research community identify much more easily the data used in each study, understand and compare data formats, and would also simplify the selection process, by making visible critical selection information [13]. Fortunately, some metadata creation can be automated; unfortunately, representation standards are also an issue for metadata.

Meta-models [38] are ontological schemes that characterize how families of different model sub-types or kinds are interrelated. Meta-models thus provide a critical framework for how to associate and integrate heterogeneous data or metadata sets into a common inter-model substrate. F/OSS tools like Protégé-2000 [43] act as meta-models editors for constructing domain-independent or domain-specific ontologies, which in turn can produce/output metadata definitions that conceptually unify different data source into common shareable views [30].

5.2 Extracting, Analyzing, Modeling, Visualizing, Simulating, Reenacting, and More with the Available Data

Tools could potentially be developed to address each of the issues reviewed in the previous section. Some such tools already partially exist in a generic form or are developed as needed by research groups. Tools such as web-scrapers that gather data, entity extractors that mine for specific entities like people and dates, or cross-references that link multiple information sources of a single project are commonly developed from scratch in each research effort. These tools are part of the basic toolbox of almost every empirical F/OSS researcher and could easily be provided as

such. In fact, several efforts are already underway to produce such tools (e.g.[27, 47]).

Another contribution of a research infrastructure could be to place research data access and manipulation tools upstream, directly within software development tools used by the F/OSS community (e.g., CVS, Subversion, Bugzilla), instead of requiring sometimes-tedious and potentially risky post processing. For example, in most cases, F/OSS tools rely on databases for data storage and manipulation. These databases contain valuable information that is often lost during the translation to a web-visible front-end. (Usually the front-ends rely on web interfaces that display information in a user-friendly fashion but drop important structure in the process). Access to the underlying database can be much more valuable (and in many cases easier) than the current techniques of web-scraping that must recreate such missing relations post-hoc, and may not be successful.

5.3 Integrated Data-to-Literature Environments: Digital libraries and new electronic journal/publishing archives featuring open content and open data.

Putting all the previous components together would lead to a set of normalized, processed and integrated collections of F/OSS data made available to the research community through either federated or centralized mechanisms. These research collections need to be curated and organized as digital libraries that organize and preserve different data sets, meta-data, derived views, analyses, and provenance that span multiple data sets/bases as well as views that span multi-disciplinary models and studies that can be accessed anytime and from anywhere [13, 54]. Furthermore, it should be both possible and desirable to offer subscription and publication services to those who want to be notified when data in the library are changed or updated [3, 12], so that they can re-analyze existing models or derived views.

Finally, an advanced contemporary approach would be an attempt to connect both data sources and research literature in a seamless and interlocking web, so that research findings can be traced back to sources, and so that basic source data can be linked directly to inferences made from it. Such arrangements provide powerful intrinsic means of discovering connections among research themes and ideas, as they are linked through both citation, through common or related uses of underlying data, and through associations among concepts. Similar efforts are underway in many other sciences (e.g., [1, 58]). Networks of literature and data created in this way, with automated support, can reduce cognitive complexity, establish collocation of concepts and findings, and establish/maintain social organization within and across F/OSS projects. The DSpace repository developed at MIT and Fedora repository [57] are among the leading research candidates that could serve as the storage and archiving facility through which F/OSS data sets, models, views, and analyses can be accessed, published, updated, and redistributed to interested researchers, in an open source, open science manner [cf. 6]. Alternatively, it may be desirable to utilize large, globally accessible repositories, index, and search services as might be provide by a commercial service provider such as Google.

6. Discussion

In our view, the multi-discipline F/OSS research community seeks to establish a *scholarly commons* that provides for communicating, sharing, and building on the ideas, artifacts, tools, and facilities of community participants in an open, globally accessible, and public way [cf. 26, 35]. A shared infrastructure, or in our case, a shared information infrastructure, is a key component and operational facility of such a commons [1, 9, 44]. Such an infrastructure provides a medium for sharing resources of common interest (e.g., F/OSS data sets, domain models, tools for processing data in F/OSS repositories, research pre-prints and publications), common-pool resources (F/OSS portals like SourceForge [55]), and public goods (scientific knowledge, Internet access and connectivity). However, a globally shared information infrastructure supporting F/OSS research may not just emerge spontaneously, though it could emerge in an *ad hoc* manner whose design and operation does not provide for a reasonably equitable distribution of access, costs, or benefits for community participants.

We want to avoid or minimize conditions that make such an infrastructure a venue for conflict (e.g., across disciplines, over data formats, making free riders pay, or rules that limit unconditional access). Consequently, community participants must allocation some attention and effort to address the infrastructure's design and operation. This in turn needs to support and embrace a diverse set of multidisciplinary research interests, but with limited resources. Unsurprisingly, this leads us to a design strategy that is not only iterative, incremental, and continuous, as many F/OSS researchers have agreed [17], but also one that embraces and builds on the practice of F/OSS development practices, processes, artifacts, and tools that are also the subject of our collective research interests. This in part seems inevitable as a way to address the concomitant need for administratively lightweight governance structures, modest and sustainable financial strategies, and national and international research partnerships among collaborators in different institutions, as well as enabling educational and community growth efforts [9, 26].

7. Recommendations

In accord with the rationales outlined above and the strong sense of the F/OSS community [17, 44], as well as from results developed at the 2008 F/OSS Repositories and Research Infrastructures Workshop (see http://fossrri.rotterdam.ics.uci.edu), we recommend that F/OSS researchers begin collective efforts to create sharable infrastructure for collaborative empirical research. This infrastructure should be assembled incrementally, with activity in many of the recommendations defined below.

This paper has provided a sketch of some ideas toward robust and useful infrastructure that can support research within and across the multiple disciplines already investing scholarly effort into the area of F/OSS. The ideas and motivations presented here however need more development, thus collaborative interdisciplinary efforts are encouraged.

Many standards for sharable scientific data exist for other communities, as do many repositories of data conforming to those standards. We should do further research on what other communities have done to organize research data. For example, many collections of social science data are maintained around the world[1]. We should use the experiences of these projects as a basis for the F/OSS research infrastructure. The success of these archives in the social science community is also a partial answer to questions of "why bother?" Beyond this, we and others recommend consideration of the following actions that may benefit the global F/OSS research community.

7.1 Instrument Existing Tools and Repositories

We should work with existing F/OSS community development tool projects to design plug-ins to instrument widely used F/OSS tools (such as Bugzilla, CVS/Subversion, etc.) to make the content of those tools available via APIs in standardized formats, administratively controllable by original tool/data owners. Such an effort could also benefit the community of F/OSS developers itself; this sort of instrumentation could help interfacing multiple tools, projects, and communities, and might increase willingness to participate. Further, finding F/OSS projects or multi-project portals that are willing to add support for wide-area event notification services [3] that can publish data set updates to remote research subscribers is real challenge that has been demonstrated to have multiple practical payoffs.

7.2 Continuously Develop Self-Managing F/OSS Research Infrastructure Prototypes

As a proof of concept, we should mock up a complete F/OSS research infrastructure model embodying as many of the desired characteristics as feasible. We should gather data sets, models, analyses, simulations, published studies and more from the many disciplines that are engaged in empirical studies of F/OSS. Such a partial implementation might use, for example, a complete cross section of sharable information from a single project, including chat, news, CVS, bug reporting, and so on. Initial efforts of this kind have produced encouraging results [15, 28, 33]. We and others have already instigated some local efforts in a few of these areas, such as generalized bug report schemas, semi-automated extraction of social processes, preliminary data taxonomies, and automated analysis tools [e.g.,14, 18, 20, 30, 32, 37, 45, 46, 47]. However, there is still a basic need to codify and capture the scientific workflow that FOSS scholars engage in when they analyze FOSS data across many repositories [22].

7.3 Federate FOSS Data Repositories

The European FOSS research community may have more resources (provided by the EC IS&T Programme) already dedicated to development of public FOSS data repositories, data analysis tools, and to the cross-sectional analysis of FOSS data, compared to the U.S. FOSS research community. At a minimum, it would benefit the U.S. FOSS research community to be able to partner with its EU and other

[1] See for example http://www.iue.it/LIB/EResources/E-data/online_archive.shtml for a list of such collections.

international research counterparts to help establish a global FOSS research infrastructure and network of federated FOSS data repositories. Beyond this, it would be of greater research value if the U.S., European, and Asian FOSS research data repositories were collectively federated within a global FOSS research data and community support infrastructure, providing resources to researchers looking to use or host repository data and for FOSS practitioners interested in providing research data.

7.4 Offer Guidance and Incentives for Contributions to a Global Census of FOSS Project Repositories

The diversity and population of FOSS project repositories is unclear and unknown. There is great interest in the research community for a baseline and ongoing census of FOSS project repositories. As FOSS projects choose to collect, organize, and share the raw data of software development as an activity in their self-interest, then it behooves us within the research community to offer some guidance or incentives for these independent FOSS projects to contribute to such a census. Similarly, we need to articulate what benefits (e.g., socio-economic impacts or intellectual contributions) the research community might offer in return to the FOSS projects that contribute to such a census.

7.5 Identify Research Questions that can Best be Answered through FOSS Data Repositories

The open and public accessibility of raw data from FOSS project repositories and multi-project repositories (e.g., SourceForge.net, FLOSSmole, Google Code [cf. 14, 15, 16, 23]) is providing a new, empirically grounded view of software technology and software development practice—a view that enables comparative, cross-sectional, and ecosystem level studies. This in turn means news kinds of research questions can be posed and new knowledge can be discovered, derived, or created. For example, repository-based studies of successful FOSS projects (of which there are now at least a few thousand such projects) indicate that their software code base, functionality, development artifacts, and developer contributions, and user base can undergo sustained exponential growth, apparently in contradiction to long-standing "laws of software evolution" which traditionally predict sub-linear, inverse square growth rates [cf. 2, 8, 34, 50]. As such, the kind of research questions that follow may ask what model or theory accounts for the super-linear evolution of FOSS systems?

Another example: are there software patterns that constitute a kind of "software genome" that characterize the evolutionary mechanisms of different families of independently developed FOSS systems? Similarly, are the critical software components or functions that lie at the heart of different software families, and does such software represent a critical evolutionary or security vulnerability (e.g., the `glibc` library is commonly bound with FOSS coded in the C programming language)? Also, what development processes best characterize FOSS projects that demonstrate sustained exponential growth of their code and functionality base, as well as the growth of the number of contributors, but with comparable

growth/decline of software quality? Last, what can we learn about the evolution of FOSS systems across multiple releases, across multiple platforms, and across different independently developed variants? Exploring any questions like these all require data drawn from multiple FOSS projects or project repositories, and this data is now available. As such, we are on the verge of possible discontinuous advance in our knowledge about software, once again based on empirical studies of FOSS.

7.6 Develop and Share Practices for Curating and Archiving FOSS Project Data

Articulating new knowledge of software products, processes, practices, and organizational forms depends in part on careful and systematic empirical study of FOSS project data. However, this data is not trivial to collect, use, or analyze. As such, there is need to articulate practices for the curation of FOSS project data in forms that increase the likelihood for the data use, reuse, and (re)analysis by people in different disciplines and settings. There is also need to help capture data provenance as well as data annotation and data analysis workflow tools & techniques. Other science disciplines have recognized similar needs, so there is an opportunity for current investments in such areas to be structured to both discipline-specific and cross-discipline research efforts. At present, the FOSS research community has little practice and access to these tools and techniques, and as a result, has little incentive to take on their application or reinvention.

7.7 Identify How Commercial Software Companies can Benefit from Studies Employing FOSS Data Repositories

The commercial software products and service industry in the U.S. is in an awkward strategic position regarding whether or how to take advantage of FOSS, or the results arising from studies of FOSS development data. Software product companies like Microsoft seem hesitant about what to do about FOSS, while software service companies like Google seem to embrace FOSS (as do computer vendors like IBM and SUN). But it appears that all software companies can benefit from empirical studies of FOSS products, processes, practices, and organizational forms that are comparative or cross-sectional, for different competitive reasons.

7.8 Encourage Corporate Sponsors of FOSS Projects to Share their Data with the Research Community

Last, companies like Google, SUN, IBM, and Microsoft Research have established a community of OSS development projects under their corporate sponsorship. These projects are sponsored as a way for these companies to help increase the pool of future software developers who might then transition into the commercial software workforce. These projects also serve to provide a situated, real-world experiment in informal software engineering education, that often takes place outside of the traditional higher education environment. However, “data” from these informal educational experiences is generally not open, nor publicly available, as it is sometimes said to be sensitive, confidential, and proprietary. Thus it is unclear how well these informal experiments work, or whether/how they can be improved both from a corporate perspective as well as from an academic perspective. Perhaps there is an opportunity to bring together the academic software research and software

engineering education community together with the commercial software industry through a government sponsored or coordinated forum so as to articulate how to best advance U.S. socio-economic and scholarly interests for mutual benefit and growth of the software community.

In the end, we believe efforts in these directions identified in these recommendations will pay off in the form of deeper collaborations within and across the empirical software research community, wider awareness of important research issues and means of addressing them, and ultimately in more systematic, grounded knowledge and theory-driven practice in software development.

Acknowledgements: Preparation of this report was supported in part through research grants from the National Science Foundation #0083705, #0205679, #0205724, #0350754, #0534771 and #0749353. No endorsement implied. Collaborators on these projects include Mark Ackerman at University of Michigan, Ann-Arbor, John Noll at Santa Clara University, Margaret Elliott, Chris Jensen, and others at the Institute for Software Research. Megan Conklin, Kevin Crowston, Greg Madey, and others also helped in organizing and contributing recommendations from the *2008 NSF Workshop of Free/Open Source Software Repositories and Research Infrastructures*, (see http://fossrri.rotterdam,ics.uci.edu) on which some of the results here are based.

References

1. Bowker G, Baker K, Millerand F. and Ribes D, (2007). Towards Information Infrastructure Studies: Ways of Knowing in a Networked Environment, in J. Hunsinger, M. Allen, and L. Klasrup (eds.), *International Handbook of Internet Research*.
2. Capiluppi A, Morisio M, and Lago, P, (2004). Evolution of Understandability in OSSProjects, *Proc. Eighth European Conf. Software Maintenance and Reengineering* (CSMR'04).
3. Carzaniga A, Rosenblum D, and Wolf A, (2001). Design and Evaluation of a Wide-Area Event Notification Service, *ACM Trans. Computer Systems*, 19(3), 332-383.
4. Choi SC, Scacchi W, (1990). Extracting and Restructuring the Design of Large Software Systems, *IEEE Software*, 7(1), 66-71, January/February.
5. Conklin M, (2007). Project Entity Matching in FLOSS Repositories, in Feller, J, Fitzgerald, B, Scacchi, W, and Sillitti, A. (Eds.), *Open Source Development, Adoption, and Innovation,* IFIP Vol. 234, Springer, 45-58.
6. David P, Spence M, (2003).*Towards an Institutional Infrastructure for E-Science: The Scope and Challenge,* Oxford Internet Institute Report, September.
7. Delorey D, Knutson C, and Chun S, (2007). Do Programming Languages Affect Productivity? A Case Study using Data from Open Source Projects,

Proc. First International Workshop on Emerging Trends in FLOSS Research and Development (*FLOSS'07*), ACM Press, Minneapolis, MN, May.

8. Deshpande A, Riehle D, (2008). The Total Growth of Open Source, *Proc. Fourth IFIP International Conference on Open Source Systems (OSS2008)*, Milan, IT (to appear, September 2008).
9. Dietz T, Ostrom E, and Stern PC, (2003). The Struggle to Govern the Commons, *Science*, 302, 1907-1912, 12 December.
10. Elliott M, Scacchi W, (2005). Free software development: Cooperation and conflict in a virtual organizational culture. In S Koch, (Ed.), *Free/Open Source Software Development*, Idea Group Publishing, Hershey, PA, 152-173.
11. English R, Schweik, C, (2007). Identifying Success and Tragedy of FLOSS Commons: A Preliminary Classification of SourceForge.net Projects, *Proc. First Intern. Workshop on Emerging Trends in FLOSS Research and Development*, Minneapolis, MN, May 2007.
12. Eugster PT, Felber PA, Guerraoui R, and Kermarrec A-M, (2003). The Many Faces of Publish/Subscribe, *ACM Computing Surveys*, 35(2), 114-131.
13. Evans, G.E, *Developing library and information center collections*. Libraries Unlimited, Englewood, CO, 4th Edition, 2000.
14. Gao Y, Madey G, (2007) Network Analysis of the SourceForge Community, in Feller, J, Fitzgerald, B, Scacchi, W, and Sillitti, A. (Eds.), *Open Source Development, Adoption, and Innovation,* IFIP Vol. 234, Springer, 187-200.
15. Garg PK, Gschwind T, and Inoue K, (2004). Multi-Project Software Engineering: An Example, *Proc. Intern Workshop on Mining Software Repositories*, Edinburgh, Scotland, May.
16. Gasser L, Ripoche G, and Sandusky R, (2004). Research Infrastructure for Empirical Science of F/OSS, Proc. Intern. Workshop on Mining Software Repositories, Edinburgh, Scotland, May.
17. Gasser L, Scacchi W, (2003). *Continuous design of free/open source software: Workshop report and research agenda*, October. http://www.isr.uci.edu/events/ContinuousDesign/Continuous-Design-OSS-report.pdf
18. German D, Mockus A, (2003). Automating the Measurement of Open Source Projects. In Proc. 3rd. Workshop Open Source Software Engineering, Portland, OR, 63-68, May.
19. GForge, (2008). Gforge Project: A Collaborative Software Development Environment, http://gforge.org. Accessed March 2008.
20. Ghosh RA, (2003). Clustering and Dependencies in Free/Open Source Software Development: Methodology and Tools, *First Monday*, 8(4), April.
21. Ghosh RA, and Prakash V, (2000). The Orbiten Free Software Survey, *First Monday,* 5(7).
22. Gil Y, Deelman E, Ellisman M, *et al*, (2007). Examining the Challenges of Scientific Workflows, *Computer*, 40(12), 24-32, December.

23. Gonzalez-Barahona JM, Lopez L, and Robles G, Community Structure of Modules in the Apache Project, Proc. 4th Intern. Workshop on Open Source Software Engineering*, 44-48,* Edinburgh, Scotland, 2004.
24. Hahsler M. and Koch S, (2005). Discussion of a Large-Scale Open Source Data Collection Methodology, *Proc. 38th Hawaii Intern. Conf. Systems Sciences*, Kailua-Kona, HI, January.
25. Hertel G, Neidner S, and Hermann S, (2003). Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel, *Research Policy*, 32(7), 1159-1177, July.
26. Hess C, Ostrom E, (2004). A Framework for Analyzing Scholarly Communication as a Commons, Presented at the *Workshop on Scholarly Communication as a Commons, Workshop in Political Theory and Policy Analysis*, Indiana University, Bloomington, IN, March 31-April 24.
27. Howison, J, Conklin M, and Crowston K, (2006). FLOSSmole: A collaborative repository for FLOSS research data and analysis. *Intern. J. Information Technology and Web Engineering*, 1(3), 17-26.
28. Huang Y, Xiang X, and Madey G, (2004). A Self Manageable Infrastructure for Supporting Web-based Simulations, *37th Annual Simulation Symposium at the Advanced Simulation Technologies Conference 2004 (ASTC'04)*, Arlington, VA, April.
29. Hunt F. and Johnson P, (2002). On the Pareto Distribution of SourceForge Projects, in C Gacekand B. Eds.), Proc. Open Source Software Development Workshop*,* 122-129, Newcastle, UK, February.
30. Jensen C, Scacchi W, (2006). Experiences in Discovering, Modeling, and Reenacting Open Source Software Development Processes, in M Li, BE Boehm, and LJ Osterweil (eds.), *Unifying the Software Process Spectrum: Proc. Software Process Workshop*, Beijing, China, May 2005, 442-469, Springer.
31. Jensen C, Scacchi, W, (2007). Role migration and advancement processes in OSSD projects: A comparative case study, in *Proc. 29th Intern. Conf. Soft. Eng,* ACM, Minneapolis, MN, 364-374.
32. Kawaguchi S, Garg PK. Inoue, K, (2003). On Automatic Categorization of Open Source Software, in Proc. 3rd Workshop OSS Engineering, Portland, OR, 63-68, May.
33. Kim S, Pan K, and Whitehead EJ, (2004). WebDAV: Open Source Collaborative Development Environment, Proc. 4th *Intern.Workshop on Open Source Software Engineering,* 44-48.
34. Koch S. (2005). Evolution of Open Source Software Systems—A Large-Scale Investigation, in *Proc. 1st Intern, Conf. Open Source Systems (OSS2005)*, Genoa, Italy.
35. Kranich N, (2004). The Role of Research Libraries in Conceptualizing and Fostering Scholarly Commons, Presented at the *Workshop on Scholarly Communication as a Commons, Workshop in Political Theory and Policy Analysis*, Indiana University, Bloomington, IN, March 31-April 2.

36. Lopez-Fernandez L, Robles G, and Gonzalez-Barahona JM, (2004). Applying Social Network Analysis to the Information in CVS Repositories, Proc. Intern Workshop on Mining Software Repositories, Edinburgh, May..
37. Madey G, Freeh V, and Tynan R, (2005). Modeling the F/OSS Community: A Quantitative Investigation, in Koch, S. (ed.), *Free/Open Source Software Development*, 203-221, Idea Group Publishing, Hershey, PA.
38. Mi P, Scacchi W (1996). A Meta-Model for Formulating Knowledge-Based Models of Software Development, *Decision Support Systems*, 17(4), 313-330.
39. Mockus A, (2007). Large-Scale Code Reuse in Open Source Software, *Proc. First International Workshop on Emerging Trends in FLOSS Research and Development* (*FLOSS'07*), ACM Press, Minneapolis, MN, May.
40. Mockus A, Fielding RT, and Herbsleb J, (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11(3):1-38, July.
41. Noll J, Scacchi W, (1991). Integrating Diverse Information Repositories: A Distributed Hypertext Approach, *Computer*, 24(12), 38-45, December.
42. Noll J, Scacchi W, (1999). Supporting Software Development in Virtual Enterprises, Journal of Digital Information, 1(4), February.
43. Noy NF, Sintek M, Decker S, Crubezy M, Fergerson RW, and Musen MA, (2001). Creating Semantic Web Contents with Protégé-2000, *IEEE Intelligent Systems*, 16(2), 60-71, March/April.
44. O'Mahony S, (2003). Guarding the Commons: How community managed software projects protect their work, *Research Policy*, 32(7), 1179-1198, July.
45. Ripoche G, Gasser L, (2003). Scalable automatic extraction of process models for understanding F/OSS bug repair. In *Proc. Intern. Conf. Software & Systems Engineering and their Applications (CSSEA'03)*, December.
46. Robles G, Gonzalez-Barahona JM, Centeno-Gonzalez J, Matellan-Olivera V, and Rodero-Merino L, (2003). Studying the evolution of libre software projects using publicly available data, in Proc. 3rd Workshop on OSS Engineering, Portland, OR, 63-68, May.
47. Robles G, Gonzalez-Barahona JM, Ghosh R, (2004). GluTheos: Automating the Retrieval and Analysis of Data from Publicly Available Software Repositories, Proc. Intern Workshop on Mining Software Repositories, Edinburgh, Scotland, May.
48. Sandusky R, Gasser L, and Ripoche G, (2004). Bug report networks: Varieties, strategies, and impacts in an OSS development community. Proc. Intern Workshop on Mining Software Repositories, Edinburgh, Scotland, May.
49. Scacchi W, (2002). Understanding the Requirements for Developing Open Source Software Systems, *IEE Proceedings--Software*, 149(1), 24-39.
50. Scacchi W, (2006). Understanding Free/Open Source Software Evolution, in N.H. Madhavji, J.F. Ramil and D. Perry (Eds.), *Software Evolution and Feedback: Theory and Practice*, John Wiley and Sons Inc, New York, 181-206.

51. Scacchi W, (2007). Free/Open Source Software Development: Recent Research Results and Methods, in M. Zelkowitz (Ed.), *Advances in Computers*, 69, 243-295.
52. Scacchi W, Jensen C, Noll J, and Elliott M, (2006). Multi-Modal Modeling, Analysis and Validation of Open Source Software Development Processes, *Intern. J. Information Technology and Web Engineering*, 1(3), 49-63, 2006.
53. Smith N, Ramil JF, Capiluppi A, (2004). Qualitative Analysis and Simulation of Open Source Software Evolution, *Proc. 5th Intern. Workshop Software Process Simulation and Modeling*, Edinburgh, Scotland, UK, 25-26 May.
54. Smith TR, (1996). The Meta-Information Environment of Digital Libraries. *D-Lib Magazine*, July/August.
55. SourceForge, (2008). http://www.sourceforge.net Accessed March 2008.
56. Sowe SK, Angelis L, Stamelos I, and Manopoulos Y, (2007). Using Repository of Repositories (RoRs) to Study the Growth of F/OSS Projects: A Meta-Analysis Research Approach, in Feller, J, Fitzgerald, B, Scacchi, W, and Sillitti, A. (Eds.), *Open Source Development, Adoption, and Innovation,* IFIP Vol. 234, Springer, 147-160.
57. Staples T, Wayland R, and Payette S, (2003). The Fedora Project: An Open-source Digital Object Repository System, *D-Lib Magazine*, April. http://www.dlib.org/dlib/april03/staples/04staples.html
58. Star SL, Ruhleder K, (1996). Steps Toward an Ecology of Infrastructure: Design and access for large information spaces. *Information Systems Research*, 7(1):111-134.
59. von Krogh G, Spaeth S, and Lakhani K, (2003). Community, joining, and specialization in open source software innovation: a case study, *Research Policy*, 32(7), 1217-1241, July.
60. Wasserman A, Capra E, (2007). Evaluating Software Engineering Processes in Commercial and Community Open Source Projects, *Proc. First Intern. Workshop on Emerging Trends in FLOSS Research and Development*, Minneapolis, MN, May.
61. West J, Gallagher S, (2006). Patterns of Open Innovation in Open Source Software, Chapter 5 in H. Chesbrough, W. Vanhaverbeke and J. West, (Eds.), *Open Innovation: Researching a New Paradigm*, Oxford, UK, Oxford University Press.