A Global Research Infrastructure for Multidisciplinary Empirical Science of Free/Open Source Software

Les Gasser[1,2], Gabriel Ripoche[1,3], Bob Sandusky[1], and Walt Scacchi[2,4]

[1]Graduate School of Library and Information Science, University of Illinois at Urbana/Champaign,
[2]Institute for Software Research, University of California Irvine,
[3]LIMSI/CNRS, Orsay, France
[4]corresponding author

Version of April 2005 (Original version, July 2004)

{gasser,gripoche,sandusky}@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 The traditional difficulties of gathering enough empirical data have been replaced by issues of dealing with enormous amounts of freely available public data from many disparate sources (online discussion forums, source code directories, bug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 common needs and critical factors in F/OSS empirical research across disciplines, and suggest orientations and recommendations for the design of a shared research infrastructure for multi-disciplinary research into F/OSS.

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 [David 2003, Gasser&Ripoche 2004, Gasser&Scacchi 2003, Hahsler and Koch 2005, Huang 2004]. 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. More than 150 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.

Objects of Study and Research Questions

As an organizing framework, we identify four 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*, *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.

Table 1: Characteristics of empirical F/OSS studies.

Objects	**Success Measures**	**Critical Driving Factors**
Artifacts	Quality, reliability, usability, durability, fit, structure, growth, Internationalization	Size, complexity, software architecture (structure, substrates, modularity, versions, infrastructure)
Processes	Efficiency, effectiveness, complexity, manageability, predictability, adaptability	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, leadership and core developers, socio-technical vision
Communities	Ease of creation, sustainability, trust, social capital, 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, applications

Current Research Approaches

We have identified at least four alternative approaches in empirical research on the objects and factors in Table 1 [cf. Gonzalez-Baharona 2004, Scacchi 2001, 2002]:

- Very large, population-scale studies examining common objects selected and extracted from hundreds to tens-of-thousands of F/OSS projects [Gao 2004, Hahsler and Koch 2005, Hunt 2002, Kawaguchi 2004, Madey 2005] or surveys of comparable numbers of F/OSS developers [Hertel 2003, Ghosh 2000]

- 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 [German 2003, Koch 2000, Smith 2004].

- Medium-scale comparative studies across multiple kinds of F/OSS projects within different communities or software system types [Capiluppi 2004, Scacchi 2002, Smith 2004]

- Smaller-scale in-depth case studies of specific F/OSS practices and processes, for concept/hypothesis

development and exposing mechanism details [Elliott 2005, Gonzalez-Baharona&Lopez 2004, Jensen 2004, Mockus 2002, O'Mahony 2003, Ripoche 2004a, Sandusky 2004, Scacchi 2004, von Krogh 2003].

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 [Gonzalez-Baharona 2004, Jensen 2004, Lopez-Fernandez 2004, Ripoche 2003b], leading to finer gradations in empirical data needs.

Essential Characteristics

Empirical studies of software artifacts, processes, communities and knowledge within and across disciplines are constrained by several key requirements. They should:

1. *Reflect actual experience* through an explicit basis or grounding, rather than assumed, artificially constructed phenomena.

2. Give *adequate coverage* of naturally-occurring phenomena, and to alternative perspectives or analytical framings.

3. Examine *representative levels of variance* in key dimensions and phenomena.

4. Demonstrate *adequate statistical significance* or *cross-cutting comparative analyses.*

5. Provide results that are *comparable across projects* within project community, or across different project communities or application domains.

6. Provide results that can be *reconstructed, tested, evaluated, extended, and redistributed* by others.

Taken together, these six requirements for multi-disciplinary F/OSS research drive several requirements on the infrastructure and data for that research. For example:

- To satisfy the needs for reality and coverage (1,2), data should be *empirical and natural*, from real projects.

- For coverage of phenomena, demonstration of variance, and statistical significance (2,3,4), data should be *available in collections of sufficient size, releases, and analytical diversity.*

- To allow for comparability across projects, and to allow community-wide testing, evaluation, extension, and redistribution of findings (5,6), data and findings should be *sharable, in common frameworks and representations*.

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), and development data (source code, bug reports, design documents, attributed file directory structures, CVS check-in logs).

Second, data originates from different *types of repository* sources [Noll 1991, Noll 1999]. These include shared file systems, communication systems, version control systems, issue tracking systems, content management systems, multi-project F/OSS portals (SourceForge.net, Freshmeat.net, Savannah.org, Advogato.org, Tigris.org, etc.),

collaborative development or project management environments [Garg 2004, GForge 2004, Kim 2004, Ohira 2004], F/OSS Web indexes or link servers (Yahoo.com/Computers_and_Internet/Software/Open_Source/, free-soft.org, LinuxLinks.com), search engines (Google), 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 F/OSS repositories is typically heterogeneous and difficult to integrate, 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 F/OSS project Web sites. Data may also be located within secondary sources appearing in research papers or paper collections (e.g., MIT F/OSS paper repository at opensource.mit.edu), where researchers have published some form of their data set within a publication.

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 gather and prepare data from one or more sources for further analysis.

Last, most F/OSS project data is available as *artifacts or byproducts* of development, usage, or maintenance activities in F/OSS communities. 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.

Issues with Empirical Data

Many steps often have to be performed to identify, gather, and prepare data before it can be used for research. Data warehousing techniques [Kimball 2002] represent a common strategy for extracting, transforming, and loading data from multiple databases into a separate single multi-dimensional database (the data warehouse) using a star schema to integrate the disparate data views. Data identification and preparation are important aspects of the research process and help guarantee that the seven essential characteristics described above are met. The following steps are common barriers that most empirical F/OSS researcher will have to address:

Discovery, Remote Sensing, and Selection

Because so much data is available, and because such diversity exists in data formats and repository types, finding and selecting pertinent, usable data to study can be difficult. This is a general Resource Description/Discovery (RDD) and information retrieval issue, appearing here in the context of scientific data. Alternatively, other approaches, rather than depending on discovery, to instead assume a proactive remote sensing scheme and mechanisms whereby data from repository "publishers" are broadcast to registered "subscribers" via topic, content or data type event notification (middleware) services [Carzaniga 2001, Eugster 2003]. Appropriate information organization, metadata, and publish/subscribe principles should ideally be employed in the original sources, but this is rare in F/OSS (and other software) data repositories, in part because of the byproduct nature of F/OSS research data.

Access, Gathering, and Extraction

By access we mean the actually obtaining useful data once it has been discovered or remotely sensed, and selected. Access difficulties include managing administrative access to data, actually procuring data (e.g., overcoming bandwidth constraints, acquiring access to remote repositories across organizational boundaries [cf. Noll 1999]), and dealing with difficulties transforming data in a useful format (such as a repository snapshot or via web scraping). Howver, when such hurdles can be overcome, then it is possible to acquire large volumes of organizational, (software) architectural configuration and version, locational, and temporal data from multiple F/OSS repositories of the same type and kind [cf. Choi 1990].

Cleaning and Normalization

Because of the diversity of research questions, styles, methods, and tools, and the diversity of data sources and repository media available, researchers face several types of difficulty with raw data from F/OSS repositories [cf. Howison 2004]: original data formats may not match research needs; data of different types, from different sources or projects, may not be easily integrated in its original forms; and data formats or media may not match those required by qualitative or quantitative data analysis tools. In these cases, research data has to be normalized before it can be used. Data normalization activities may include data format changes, integration of representation schemas, transformations of basic measurement units, and even pre-computation and derivation of higher-order data values from base data. Normalization issues appear at the level of individual data items and at the level data collections.

Clustering, Classifying, and Linked Aggregation

Normalized data is critical for cross-source comparison and mining over data "joins". However, some F/OSS-based research projects are exploring structural links and inferential relationships between data of very different characters, such as linking social network patterns to code structure patterns [Gonzalez-Baharona&Lopez 2004, Lopez-Fernandez 2004, Madey 2005], or linking bug report relationships to forms of social order [Sandusky 2004]. Linked data aggregation demands invention of new representational concepts or ontological schemes specific to the kinds of data links desired for projects, and transformations of base data into forms compatible with those links. Whether these representations are automatically constructed through bottom-up data-driven approaches [Ripoche 2003b], top-down model-driven approaches [Jensen 2004], or some hybrid combination of both together with other machine learning techniques, remains an open topic for further investigation

Integration and Mobilization

Data from heterogeneous sources must be integrated into homogeneous views for further processing and rendering. Techniques including wrappers, brokers, or gateways are used as middleware techniques for selecting and translating from locally heterogeneous source specific data forms into homogeneous forms that can be integrated into global views [Noll 1999]. Such an integration scheme allows for different types and kinds of data views to be constructed in ways that maintain the autonomy of data sources, while enabling transparent access by different types of clients (users or data manipulation tools) to remote data sources [Noll 1991]. Finally, it enables the mobility of views across locations so that research users in different geographic and institutional locations can access data from common views, as if the data were located in-house, even though each such access location may utilize its own set of middleware intermediaries. Thus, these views can serve as virtual data sets that appear to be centrally located and homogeneous, but are physically decentralized and heterogeneous.

Evolution

Real projects continually evolve, both in content and in format: web sites are redesigned, tools are modified, etc. Research projects may have to track, adapt to, and reflect these changes. This can cause problems at many of the previous levels, as access rights can be modified, formats can change and links can be created or removed. In addition, trajectories of evolution themselves are actually an important object of study for some empirical software researchers. The central issue for this paper is how to adhere to the essential characteristics given above (such as the needs for testable, repeatable, and comparable results) while reacting to and/or managing this evolution.

Addressing These Issues

The main objective of a research infrastructure is to address community-wide resource issues in community-specific way [Star 1996]. 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.

Representation Standards for Data, Processes, Protocols

One of the broadest approaches to common infrastructure is the use of representation standards [Star 1996]. Such standards would move some issues of cross-source data normalization forward in the process that produces F/OSS projects' information. For example, standard internal formats for objects such as bug reports could eliminate many internal differences between Bugzilla, Scarab, Gnats, etc., fostering simpler cross-analysis of data from these various repositories. Such representation standards would also facilitate exchange of data and/or processing tools within the F/OSS research community. For example, as part of one investigation of F/OSS bug reporting/resolution processes [Ripoche 2003b], we developed a general XML schematization of bug reports, derived from (but more general than) the Bugzilla internal database schema, and designed as a normalization target and translation medium for multiple types of bug reports from different systems [Ripoche 2003a]. Issues include the difficulty of developing, promulgating, maintaining, and enforcing such standards.

Metadata and Meta-models

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 [Smith 1996]. 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. Fortunately, some metadata creation can be automated; unfortunately, representation standards are also an issue for metadata.

Meta-models [Mi 1996, Scacchi 1998] are ontological schemes that characterize how families of different 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 [Noy 20001] 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 [Jensen 2004, Scacchi 1998].

Logically Centralized but Physically Decentralized Virtual Data Repositories (VDRs)

Gathering specific snapshots of raw data and making them available to the research community from a controlled “cleanroom” location could provide benchmark data for comparative analyses and measurement of progress--a type of infrastructure that has proven invaluable in other disciplines [cf. Noll and Scacchi 1999]. It would ensure that data parameters stay constant across studies, and through evolutionary stages of projects. Moreover, it might be easier in many cases to get a snapshot from such a repository than to go through all the steps of collecting the data directly from an F/OSS community. The VDR approach can have advantages of control, organization, and data persistence. However, this approach also raises the issues of *data selection* and *maintenance*. As with any managed information collection, VDRs would need *selection policies* to detail which materials from projects, tools and communities would be chosen for inclusion, and why [Evans 2000]. The F/OSS community is already too large to attempt building practical evolving archives of *all* F/OSS projects (if such a notion were even meaningful). Selection necessarily induces bias, but careful selection would foster research on a shared body of data, possibly leading to more reliable findings. Second, *preservation policies* need development as F/OSS data is evolving quickly and collections will have to be maintained.

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 nascent efforts are already underway to produce such tools (e.g. [Libre 2004]).

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.

Federated Access and Replicated Infrastructure Mirrors

Federating access is another approach to facilitating information sharing without making many redundant copies of original data, while maintaining local control over data access and organization. A central federation repository collects only metadata, and uses it to provide common-framework access to a variety of underlying sources. Federation has the advantages of distributed sharing, such as trading off lightweight central representations and sophisticated search infrastructure, against local data maintenance, information preservation, and access control.

Processed Research Collections

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 organized as digital libraries that organize and preserve different data sets, meta-data, and derived views that span multiple data sets/bases as well as views that span multi-disciplinary models and analyses that can be accessed anytime and from anywhere [Smith 1996]. 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 [Eugster 2003], so that they can re-analyze existing models or derived views.

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

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. [Shoman 1995, Star 1996]). 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 [2004] repository developed at MIT and Fedora repository [Staples 2003] are among the leading 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. David 2003].

Discussion

In our view, the multidiscipline 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. Hess 2004, Kranich 2004]. A shared infrastructure, or in our case, a shared information infrastructure, is a key component and operational facility of such a commons [Dietz 2003]. 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 [2004]), 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, some attention and effort must be allocated by community participants to begin to address the infrastructure's design and operation, so as 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 [Gasser&Scacchi 2003], 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 [Dietz 2003, Hess 2004].

Recommendations

In accord with the rationales outlined above and the strong sense of the F/OSS community [Gasser&Scacchi 2003], 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 areas defined below:

Refine Knowledge

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.

Exploit Experience

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?"

Instrument Existing Tools and Repositories

We should work with existing F/OSS community development tool projects to design plugins for instrumenting 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 that can publish data set updates to remote research subscribers is real challenge that has been demonstrated to have multiple practical payoffs [Gard 2004, Huang 2004, Ohira 2004]

Develop Data Standards

Standards for metadata and representation will help glue together data and tools such as finding aids and normalization tools. In collaboration with F/OSS tool developers, we should work toward standardizing formats and

---

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

content of repositories of many kinds.

Create Federation Middleware

Federated approaches to data archives will have much lower initial costs and will foster community building while maintaining local control over base data and sharing. The foundation for such middleware exists, as can be found in Digital Library frameworks such as Fedora [Staples 2003].

Develop Consensus on Data Selection Policies

We need much more consensus on what kinds of data provide the most utility for the widest variety of empirical F/OSS research projects. Developing this consensus will also help to congeal the community of empirical software researchers.

Create and Continuously Design 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 [Garg 2004, Huang 2004, Kim 2004, Ohira 2004]. We 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, automated analysis tools, and others have also begun efforts in these directions [e.g., Jensen 2004, Gao 2004, German 2003, Ghosh 2003, Kawaguchi 2003, Libre 2004, Madey 2004, Ohira 2004, Ripoche 2003a,b, Robles&Gonzalez-Baharona 2003, Robles, Gonzalez-Baharona & Ghosh 2004].

In the end, efforts in these directions 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 and #0350754. 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, Richard Taylor, and others at the Institute for Software Research.

Bibliography

Alonso, O, Devanbu, P.T., and Gertz, M., Database Techniques for the Analysis and Exploration of Software Repositories, *Proc. Intern Workshop on Mining Software Repositories*, Edinburgh, Scotland, May 2004.

Capiluppi, A., Morisio, M., and Lago, P., Evolution of Understandability in OSS Projects, *Proc. Eighth European Conf. Software Maintenance and Reengineering* (CSMR'04), 2004.

Carzaniga, A., Rosenblum, D. and Wolf, A., Design and Evaluation of a Wide-Area Event Notification Service, *ACM Trans. Computer Systems*, 19(3), 332-383, 2001.

Choi, S.C. and Scacchi, W., Extracting and Restructuring the Design of Large Software Systems, *IEEE Software*, 7(1), 66-71, January/February 1990.

*** Creative Commons, http://www.creativecommons.org, 2005.

David, P. and Spence, M., *Towards an Institutional Infrastructure for E-Science: The Scope and Challenge*, Oxford Internet Institute Report, September 2003.

Dietz, T., Ostrom, E., and Stern, P.C., The Struggle to Govern the Commons, *Science*, 302, 1907-1912, 12 December 2003.

Elliott, M. and Scacchi, W., 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, 2005.

EPrints, GNU EPrints Archive Software, http://software.eprints.org/ August 2004.

Evans, G.E., *Developing library and information center collections*. Libraries Unlimited, Englewood, CO, 4th Edition, 2000.

Eugster, P.T., Felber, P.A., Guerraoui, R., and Kermarrec,, A-M., The Many Faces of Publish/Subscribe, *ACM Computing Surveys*, 35(2), 114-131, June 2003.

Fenton, A., Software Measurement: A Necessary Scientific Basis, *IEEE Trans. Software Engineering*, 20(3), 199-206, 1994.

Gao, Y., Huang, Y., and Madey, G., Data Mining Project History in Open Source Software Communities, *NAACSOS Conference 2004*, Pittsburgh, PA, June 2004

Garg, P.J., Gschwind, T., and Inoue, K., Multi-Project Software Engineering: An Example, *Proc. Intern Workshop on Mining Software Repositories*, Edinburgh, Scotland, May 2004.

Gasser, L., Ripoche, G. and Sandusky, R., Research Infrastructure for Empirical Science of F/OSS, *Proc. Intern. Workshop on Mining Software Repositories*, Edinburgh, Scotland, May 2004.

Gasser, L. and Scacchi, W., *Continuous design of free/open source software: Workshop report and research agenda*, October 2003. http://www.isr.uci.edu/events/ContinuousDesign/Continuous-Design-OSS-report.pdf .

German, D. and Mockus, A., Automating the Measurement of Open Source Projects. In Proc. 3rd. Workshop Open Source Software Engineering, Portland, OR, 63-68, May 2003.

GForge Project: A Collaborative Software Development Environment, http://gforge.org, August 2004.

Ghosh, R., Clustering and Dependencies in Free/Open Source Software Development: Methodology and Tools, *First Monday*, 8(4), April 2003.

Ghosh, R.A. and Ved Prakash, V., The Orbiten Free Software Survey, *First Monday*, 5(7), July 2000.

Gonzalez-Barahona, J.M., 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.

Gonzalez-Barahona, J.M. and Robles, G., Getting the Global Picture, Presentation at the *Oxford Workshop on Libre Software* (OWLS), Oxford Internet Institute, Oxford, England, 25-26 June 2004.

Jensen, C. and Scacchi, W., Data Mining for Software Process Discovery in Open Source Software Development Communities, *Proc. Intern Workshop on Mining Software Repositories*, Edinburgh, Scotland, May 2004.

Hahsler, M. and Koch, S., Discussion of a Large-Scale Open Source Data Collection Methodology, *Proc. 38th Hawaii Intern. Conf. Systems Sciences*, Kailua-Kona, HI, Jan 2005.

Hertel, G., Neidner, S., and Hermann, S., 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 2003.

Hess, C., and 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 2, 2004.

Howison, J. and Crowston, K., The perils and pitfalls of mining SourceForge, *Proc. Intern Workshop on Mining Software Repositories*, Edinburgh, Scotland, May 2004.

Huang, Y., Xiang, X., and Madey, G., 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 2004. (paper)

Hunt, F. and Johnson, P., On the Pareto Distribution of SourceForge Projects, in C. Gacek and B. Arief (Eds.), Proc. Open Source Software Development Workshop, 122-129, Newcastle, UK, February 2002.

Kawaguchi, S., Garg, P.K., Matsushita, M., and Inoue, K., On Automatic Categorization of Open Source Software, in Proc. 3rd Workshop OSS Engineering, Portland, OR, 63-68, May 2003.

Kim, S., Pan, K., and Whitehead, E.J., WebDAV Open Source Collaborative Development Environment, *Proc. 4th Intern.Workshop on Open Source Software Engineering*, 44-48, Edinburgh, Scotland, 2004,

Kimball, R. and Ross, M., *The Data Warehouse Toolkit, Second Edition*, Wiley, New York, 2002.

Koch, S. (Ed.), *Free/Open Source Software Development*, Idea Group Publishing, Hershey, PA, 2005.

Koch, S. and Schneider, G., Results from software engineering research into open source development projects using public data. Diskussionspapiere zum Tätigkeitsfeld Informationsverarbeitung und Informationswirtschaft, H.R. Hansen und W.H. Janko (Hrsg.), Nr. 22, Wirtschaftsuniversität Wien, 2000.

Kranich, N., 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, 2004.

*** Kuro5hin, http://www.kuro5hin.org, 2005.

Libre Software Engineering tool repository. http://barba.dat.escet.urjc.es/index.php?menu=Tools, August 2004.

Lopez-Fernandez, L., Robles, G., and Gonzalez-Barahona, J.M., Applying Social Network Analysis to the Information in CVS Repositories, *Proc. Intern Workshop on Mining Software Repositories*, Edinburgh, May 2004.

Madey, G., Freeh, V., and Tynan, R., Modeling the F/OSS Community: A Quantitative Investigation," in Koch, S. (ed.), *Free/Open Source Software Development*, 203-221, Idea Group Publishing, Hershey, PA, 2005.

Mi, P. and Scacchi, A Meta-Model for Formulating Knowledge-Based Models of Software Development, *Decision Support Systems*, 17(4):313-330, 1996.

Mockus, A., Fielding, R.T., and Herbsleb, J., Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11(3):1-38, July 2002.

NRC, National Research Council, *Bits of Power: Issues in Global Access to Scientific Data*, National Academies Press, 1997, http://www.nap.edu/readingroom/books/BitsOfPower.

Noll, J. and Scacchi, W., Integrating Diverse Information Repositories: A Distributed Hypertext Approach, *Computer*, 24(12):38-45, December 1991.

Noll, J. and Scacchi, W., Supporting Software Development in Virtual Enterprises, Journal of Digital Information, 1(4), February 1999.

Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W., and Musen, M.A., Creating Semantic Web Contents with Protégé-2000, *IEEE Intelligent Systems*, 16(2), 60-71, March/April 2001.

OAI, Open Archives Initiative, http://www.openarchives.org/ August 2004.

Ohira, M., Yokomori, R., Sakai, M., Matsumoto, K., Inoue, K., and Torii, K., Empirical Project Monitor: A Tool for Mining Multiple Project Data, *Proc. Intern Workshop on Mining Software Repositories*, Edinburgh, May 2004.

O'Mahony, S., Guarding the Commons: How community managed software projects protect their work, *Research Policy*, 32(7), 1179-1198, July 2003.

*** Parker, G. and Van Alstyne, M., Mechanism Design to Promote Free Market and Open Source Software Innovation, *Proc. Hawaii Intern. Conf. Systems Sciences, Kailua-Kona*, HI, January 2005.

Reichman, J. and Uhlir, P., A Contractually Reconstructed Research Commons for Scientific Data in a Highly Protectionist Intellectual Property Environment, *Law & Contemp. Probs*. 66(1/2), 315-462, Winter/Spring 2003..

Ripoche, G. and Gasser, L., Possible Bugzilla Modification to Create a Web-API for Direct XML Serialization of Bug Reports. SQA Project Memo UIUC-2003-20, 2003a.

Ripoche, G. and Gasser, L., Scalable automatic extraction of process models for understanding F/OSS bug repair. In *Proc.Intern.Conf. Software & Systems Engineering and their Applications (CSSEA'03)*, Paris, France, December 2003b.

Robles, G., Gonzalez-Barahona, J.M., Centeno-Gonzalez, J., Matellan-Olivera, V., and Rodero-Merino, L., Studying the evolution of libre software projects using publicly available data, in Proc. 3rd Workshop on OSS Engineering, Portland, OR, 63-68, May 2003.

Robles, G., Gonzalez-Barahona, J.M., Ghosh, R., GluTheos: Automating the Retrieval and Analysis of Data from Publicly Available Software Repositories, *Proc. Intern Workshop on Mining Software Repositories*, Edinburgh, Scotland, May 2004.

Sandusky, R., Gasser, L., and Ripoche, G., Bug report networks: Varieties, strategies, and impacts in an OSS development community. *Proc. Intern Workshop on Mining Software Repositories*, Edinburgh, Scotland, May 2004.

Scacchi, W., Modeling, Integrating, and Enacting Complex Organizational Processes, in Carley, K., Gasser, L., and Prietula, M., (eds.), *Simulating Organizations: Computational Models of Institutions and Groups*, 153-168, MIT Press, 1998.

Scacchi, W., Software Development Practices in Open Software Development Communities, 1st Workshop on Open Source Software Engineering, Toronto, Ontario, May 2001.

Scacchi, W., Understanding the Requirements for Developing Open Source Software Systems, *IEE Proceedings--Software*, 149(1), 24-39, February 2002.

Shoman, L., Grossman, E., Powell, K., Jamison, C., and Schatz, B., The Worm Community System, release 2.0 (WCSr2). *Methods in Cell Biology*, 48:607-625, 1995.

Smith, N., Ramil, J.F., Capiluppi, A., Qualitative Analysis and Simulation of Open Source Software Evolution, *Proc. 5th Intern. Workshop Software Process Simulation and Modeling*, Edinburgh, Scotland, UK, 25-26 May 2004.

Smith, T.R., The Meta-Information Environment of Digital Libraries. *D-Lib Magazine*, July/August 1996.

SourceForge Open Source Software Development Website, http://sourceforge.net, August 2004.

Staples, T., Wayland, R., and Payette, S., The Fedora Project: An Open-source Digital Object Repository System, *D-Lib Magazine*, April 2003. http://www.dlib.org/dlib/april03/staples/04staples.html

Star, S.L. and Ruhleder, K., Steps Toward an Ecology of Infrastructure: Design and access for large information spaces. *Information Systems Research*, 7(1):111-134, 1996.

von Krogh, G., Spaeth, S., and Lakhani, K., Community, joining, and specialization in open source software innovation: a case study, *Research Policy*, 32(7), 1217-1241, July 2003.