Issues and Experiences in Modeling Open Source Software Development Processes

Walt Scacchi
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3425 USA
949-824-4130, 949-824-1715 (fax)
Wscacchi@uci.edu
http://www.ics.uci.edu/~wscacchi

Overview

This study presents selected analyses and findings from a multi-year study into the development processes, work practices, and community dynamics that arise in open source software development (OSSD) projects. Previous results from this study have identified socio-technical development processes that shape OSSD projects [Scacchi 2001]; the use of software informalisms as both OSSD artifacts and communication media in developing the requirements for OSSD projects [Scacchi 2002a]; investigation of the comparative advantages that arise in OSSD versus traditional software engineering [Scacchi 2002b]; and others [Elliott and Scacchi 2003, Scacchi 2002c].

In this paper, emphasis is directed to presenting preliminary findings as to the kinds issues that arise in the modeling of the techno-social processes found in different OSSD projects. We highlight results from the examination, discovery, and modeling of software engineering processes [Scacchi 2002d] within the Apache HTTP (Web) server, Mozilla Web browser, and in particular, the NetBeans interactive development environment (IDE) project.

It is important to qualify the findings reported here in ways that have previously lacked adequate attention. In particular, it is important to recognize that efforts like Apache and Mozilla are *composite projects* that consist of many loosely-coupled component system development projects. For example, the Apache Software Foundation consists of a dozen or so top level projects such as the HTTP Server, ANT, Cocoon, and Jakarta, while each of these contains their own sub-projects. For instance, the Jakarta project, focusing on the creation of commercial-quality, server-side solutions using Java, contains about two dozen sub-projects, such as Tomcat and Lucene. The Apache HTTP Server also has multiple system component sub-projects. Therefore, what is observed and reported as findings in one sub-project may not be indicative of the practices or processes found in other sub-projects within its parent projects, or the overall composite project. Such a condition therefore merits consideration when examining the results from studies of large composite projects that have been published elsewhere, so

as to appropriately gauge or qualify the scope of the reported findings.

In the remainder of this paper, attention is focused to issues and findings from an effort to empirically discover and model software engineering processes that can be observed arising in particular OSSD. In this case, the choice is to examine issues arising in the NetBeans IDE projects. NetBeans is more of a large-scale (monolithic) OSSD project that we find in many ways resembles traditional distributed software engineering projects, compared to common small-scale OSSD projects, or to the composite OSSD projects like Apache, Mozilla, and Linux.

OSSD Process Discovery

Previous studies that report on OSSD projects like Apache Web server and Mozilla Web browser [Mockus 2002, Fielding, and Herbsleb 2002, Reis and Fortes 2002] provide informal narrative descriptions of the overall software development process that typify how these systems are developed. However, they do not treat these processes in terms that associate tools or the roles that people play in performing different development tasks. Similarly, they do not identify which project or Web site repositories are employed to store the artifacts that document or embody these systems [cf. Scacchi 2002a]. Finally, neither Apache, Mozilla, NetBeans, or any other OSSD projects that have been examined provide documents on their project Web sites that explicitly describe what development processes are employed in the projects, nor how they are to be performed. As a result, any software engineer or developer seeking to either join one of these projects, or to start an independent OSSD project, cannot find explicit descriptions or models of the software processes that are performed to support the production of OSS systems.

Instead, if one seeks to understand and perform OSSD processes, then these processes must be discovered either through ad hoc trial-and-error, or by systematic investigation, Web site navigation, coding, and modeling. The former choice offers no leverage, while the latter implies labor-intensive research methodology, such as virtual ethnography [Hine 2001, Scacchi 2002a, Viller and Sommerville 2001]. The latter approach is taken here.

To capture and code the results from such an inquiry, it is necessary to organize and associate (e.g., hyperlink) observed development roles, tools, and development tasks. One such method for doing this is to employ a *rich picture* [Monk and Howard 1998] to visualize the observed associations in the context of a specific project. In the rich picture of the NetBeans OSSD project shown in Figure 1, project participants play roles including Users, CVS administrator, Web site administrator, the Release Manager, The Board, Sun Microsystems, etc. Associated with each participant role are their stakeholder concerns or interests (indicated in the bubble clouds), development activities (download release, report bugs, etc.), tools (CVS, etc.) and associated artifacts (software version release, bug reports, etc.) as hyperlinked Use Cases (not shown).

Together, the rich picture and the Use Cases serve to capture and depict contextual features of the NetBeans development effort in terms of the processes, concerns, and overall structure of relations that serve as a prelude to construction of a formal model. Aspects of the resulting model are described in the next section.

Beyond this, a companion effort describes recent progress in the development of automated mechanisms to support and streamline this process discovery effort [Jensen and Scacchi, 2003].

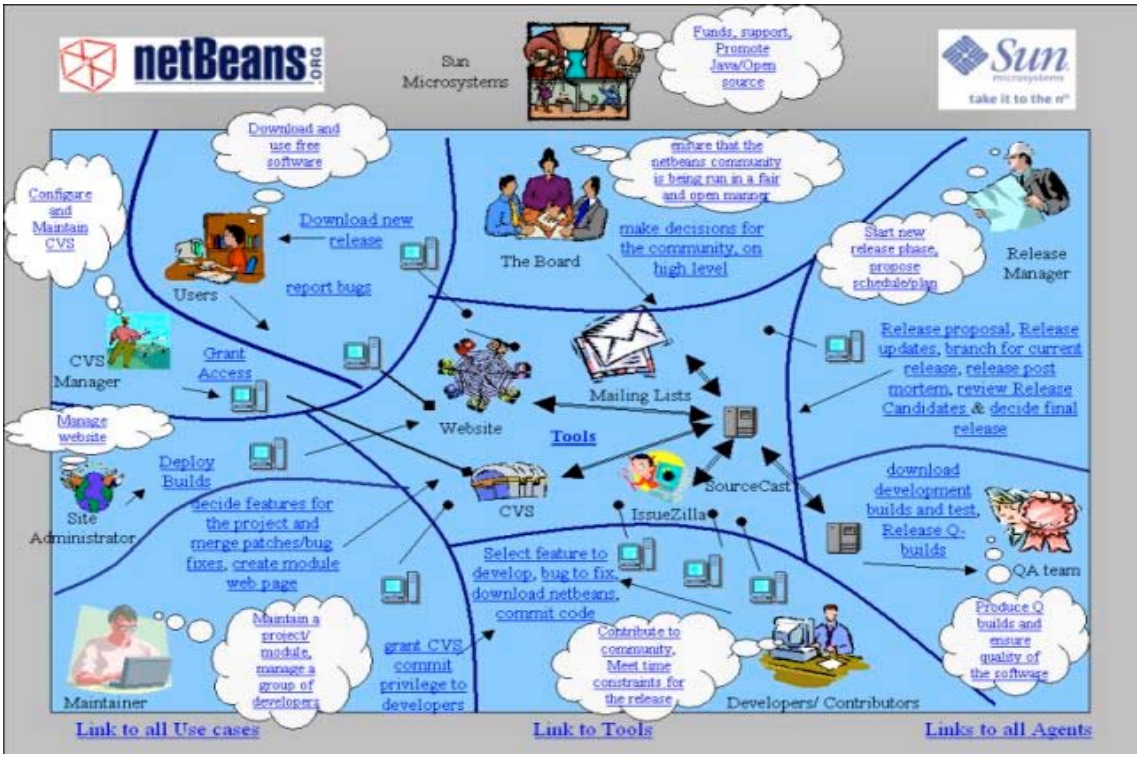

Figure 1. A rich picture depicting relationships between developer (agent) roles, artifacts, and tools associated with the software release process observed in the NetBeans project.

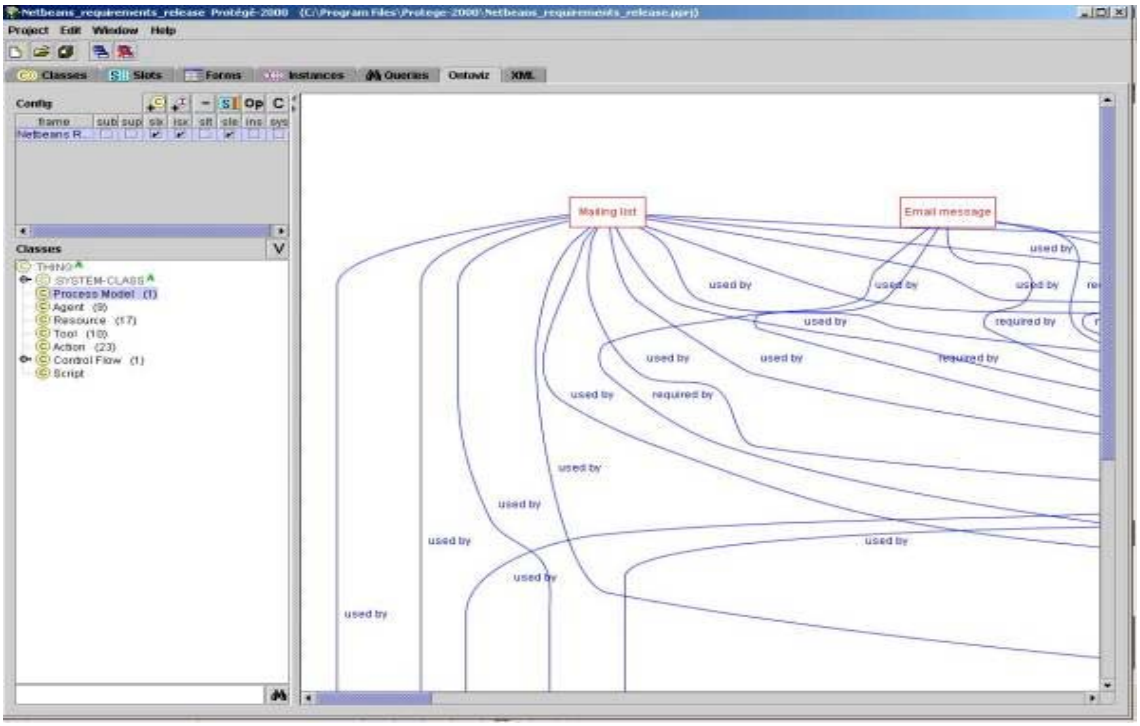

Figure 2. A partial view of a Protégé-based model of objects (artifacts, tools, roles) and activities (links) that form a formal model of the software release process for the NetBeans OSSD project.

OSSD Process Modeling

The available studies of Apache and Mozilla projects noted above also do not provide either a visual flow or a formal computational model of these processes. As such, the available narrative descriptions of these processes can not be easily analyzed, compared, visualized, computationally enacted, or transferred for (re)use in other projects [cf. Noll and Scacchi 2001, Scacchi 2002d].

The goal of this study is to discover and codify empirically observed OSSD processes as formal, computationally enactable models using the PML process modeling language [Noll and Scacchi 2001]. Software processes modeled in PML employ the ontology of a process meta-model as their semantic foundation [Mi and Scacchi 1996], and this foundation has been installed and configured to operate within a semantic object modeling framework called Protégé [Noy, *et al*., 2001]. Figure 2 displays a partial view of a formal model of the software release process within the NetBeans OSSD project that conforms to this process meta-model [Georgas 2002, Oza, *et al.,* 2002].

Among other things, Protégé can be programmed to capture (manually), edit, and transform models of software processes into notational forms like PML, as well as XML and SQL. Further, it should be noted that Protégé is itself an OSS tool for creating knowledge base models and ontologies.

Overall, one of the principal goals for modeling software engineering processes is to establish informal narrative, semi-structured hypermedia (i.e., a Web-based rich picture image map), and formal computational renderings that can be analyzed, compared, and shared within both the research and practice communities. Modeling also provides the foundation for continuous process improvement. The OSSD community has up to this time not recognize the potential value or use of "open source" software process models. As a result, OSSD projects like NetBeans must rediscover or reinvent software engineering processes, rather than follow the practice of open sharing, modification (improvement), and redistribution of these processes in ways that should enhance the productivity and quality of OSSD projects. The effort introduced in this study perhaps marks a first step in this direction.

Conclusions

This paper introduces and examines some of the issues that arise when seeking to discover and computationally model the software engineering processes that arise in a sample of different open source software development projects. Capturing and modeling of these processes may help developers new to an established composite OSSD project, or starting a new OSSD project, to more rapidly learn and become productive in the enactment of these processes. The discovery and modeling of these processes can also be an effective enabler of continuous process improvement techniques, as well as serve as another coordination resource that globally distributed developers can employ to stabilize and synchronize their loosely couple software development activities, roles and artifacts.

Acknowledgements

The research described in this report is supported by grants from the National Science Foundation #IIS-0083075 and #ITR-0205679. No endorsement implied. Contributors to work described in this paper include John Georgas, who tailored the Protégé tool for use in software process modeling; Mark Ackerman at the University of Michigan Ann Arbor; Les Gasser at the University of Illinois, Urbana-Champaign; John Noll at Santa Clara University; Margaret Elliott, Chris Jensen, Mark Bergman, and Xiaobin Li at the UCI Institute for Software Research; and Julia

Watson at The Ohio State University are also collaborators on the research project described in this paper.

References

ELLIOTT, M. and SCACCHI, W.: Free Software Development: Cooperation and Conflict in A Virtual Organizational Culture, to appear in S. Koch (ed.), *Free and Open Source Software Development*, IDEA Press, 2003.

GEORGAS, J.: Software Process Modeling with Protégé, http://www.ics.uci.edu/~jgeorgas/ics225/, May 2002.

HINE, C.: *Virtual Ethnography*, SAGE Publishers, London, 2000.

JENSEN, C. and SCACCHI, W.: Automating the Discovery and Modeling of Open Source Software Development Processes, submitted to the *3rd Workshop on Open Source Software Engineering*, Portland, OR, 2003.

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

MOCKUS, A., FIELDING, R. and HERBSLEB, J.: Two Case Studies on Open Source Software Development: Apache and Mozilla, *ACM Trans. Software Engineering and Methodology,* **11**(3), 309-346, 2002.

MONK, A. and HOWARD, S.: The Rich Picture: A Tool for Reasoning about Work Context, *Interactions*, 21-30, March-April 1998.

NOLL, J. and SCACCHI, W.: Supporting Software Development in Virtual Enterprises, *J. Digital Information*, **1**(4), February 1999.

NOLL, J. and SCACCHI, W.: Specifying Process-Oriented Hypertext for Organizational Computing, *J. Network and Computer Applications*, **24**(1), pp. 39-61, 2001.

NOY, N.F., SINTEK, M., DECKER, S., CRUBEZY, M., FERGERSON, and R.W. MUSEN, M.A.: Creating Semantic Web contents with Protege-2000, *IEEE Intelligent Systems*, **16**(2), 60-71, Mar/Apr 2001.

OZA, M., NISTOR, E., HU, S., JENSEN, C., and SCACCHI, W.: A First Look at the Netbeans Requirements and Release Process, http://www.ics.uci.edu/~cjensen/papers/FirstLookNetBeans/, June 2002.

REIS, C.R. and FORTES, R.P.M.: An Overview of the Software Engineering Process and Tools in the Mozilla Project, *Proc. Workshop on Open Source Software Development*, Newcastle, UK, February 2002.

SCACCHI, W.: Software Development Practices in Open Software Development Communities, *Proc. 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, 2002a.

SCACCHI, W.: Is Open Source Software Development *Faster*, *Better*, and *Cheaper* than Software Engineering? *Proc. 2nd Workshop on Open Source Software Engineering*, Orlando, Fl, May 2002b.

SCACCHI, W.: *Open EC/B: A Case Study in Electronic Commerce and Open Source Software Development*, technical report, Institute for Software Research, UC Irvine, July 2002c.

SCACCHI, W.: Process Models in Software Engineering, in J. Marciniak (ed.), *Encyclopedia of Software Engineering, Second Edition*, 993-1005, Wiley, 2002d.

VILLER, S. and SOMMERVILLE, I.: Ethnographically Informed Analysis for Software Engineers, *Int. J. Human-Computer Studies*, 53, 169-196, 2001.