Collaboration, Leadership, Control, and Conflict Negotiation in the *Netbeans.org* Open Source Software Development Community

Chris Jensen and Walt Scacchi
Institute for Software Research
Donald Bren School of Information and Computer Sciences
University of California, Irvine
Irvine, CA, USA 92697-3425
{cjensen, wscacchi}@ics.uci.edu

Abstract

Large open source software development communities are quickly learning that, to be successful, they must integrate efforts not only among the organizations investing developers within the community and unaffiliated volunteer contributors, but also negotiate relationships with external groups hoping to sway the social and technical direction of the community and its products. Leadership and control sharing across organizations and individuals in and between communities are common sources of conflict. Such conflict often leads to breakdowns in collaboration. This paper seeks to explore the negotiation of these conflicts, collaborative efforts, and leadership and control structures in the Netbeans.org community.

Keywords

Collaboration, Conflict Negotiation, Leadership, Process, Open Source Software Development, Netbeans.org

1. Introduction

Is open source software development (OSSD) best characterized as being strictly cooperative, or as cooperative and in conflict at the same time [Easterbrook 1993]? Conflict clearly arises during sustained software development efforts [e.g., Sawyer 2001]. But previous studies of conflict associated with Internet-based communities has focused attention to that found in specific OSSD projects operating as virtual organizations [Elliott and Scacchi 2003], as non-profit foundations [O'Mahony 2004], or in online discussion communities [Smith 1999]. None of these studies specifically help us understand the kinds of conflict, cooperation, and collaboration that arises or is needed to coordinate large-scale OSSD processes and effort in large project communities where corporate sponsorship may be a central facet of OSSD.

NetBeans.org is one of the largest OSSD communities around these days [cf. Jensen and Scacchi 2003]. Netbeans.org is a Java-focused OSSD community backed by Sun Microsystems devoted to creating both an integrated development environment (IDE) for developing large Java-based applications, as well as a platform for development of other software products. Originally started as a student project in 1996, the Netbeans.org project was acquired and subsequently released as an open source community project by Sun, whose Netbeans.org team includes many of the community's core developers. While the issues presented here stem from observations in the Netbeans.org community, they are by no means limited to this community, nor have their challenges been insurmountable.

Our study focuses on three items. First, we identify the objects of interaction among participants in the NetBeans.org community that are media through which collaboration, leadership, control and conflict negotiation are expressed and enacted. Second, we explore relationships arising in NetBeans.org on an intra-community level. Then, we look at relationships between communities like Netbeans.org and other communities and organizations.

2. Objects of Interaction

Much of the development work that occurs in an open source software project centers around the creation, update, and other actions (e.g., copy, move, delete) applied to a variety of software development artifacts. These artifacts serve as coordination mechanisms [Schmidt and Simone 1996, Simone and Mark 1999], in that they help participants communicate, document, and otherwise make sense of what the emerging software system is suppose to do, how it should be or was accomplished, who did what, what went wrong before, how to fix it, and so forth. Furthermore, within a project community these artifacts help coordinate local, project-specific development activities, whereas between multiple project communities, these artifacts emerge as boundary objects [Star 1990] through which inter-community activities and relations are negotiated and revised. The artifacts may take the form of text messages posted to a project discussion list, Web pages, source code directories and files, site maps, and more, and they are employed as the primary media through which software requirements and design are expressed. These "software informalisms" [Scacchi 2002] are especially important as coordination mechanisms in OSSD projects since participants generally are not co-located, they do not meet face-to-face, and authority and expertise relationships among participants is up for grabs.

The NetBeans IDE is intended to support the development of Web-compatible Java applications. In the context of the NetBeans.org project and its role within a larger Web-compatible information infrastructure, additional artifacts come into play within and across projects. These include the content transfer protocols like the HyperText Transfer Protocol (http) which are systematically specified in Internet standards like RFC documents, as well as more narrowly focused communication state controllers associated with remote procedure calls (or remote method invocations). They also include shared data description formats like the HyperText Markup Language (html) and the eXtensible Markup Language (XML), as well as client-side or server-side data processing scripts (e.g., CGI routines). Such descriptions may be further interpreted to enable externally developed modules to serve as application/module plug-ins, which enable secondary or embedded applications to be associated with an OSS system. Other artifacts are brought in from other OSSD projects to serve as project support tools, such as those used to record and store system defect (bug) reports (Issuzilla), email list managers, and even large comprehensive collaborative software development environments and project portals, like SourceCast [Augustin, Bressler, and Smith 2002]. Finally, OSSD projects may share both static and operational artifacts in the course of collaborating or cooperating through mutually intelligible and interoperable development processes, which might take an explicit form like the Java Community Process (JCP), or an implicit and embedded form such as that which emerges from use of project repositories whose contents are shared and synchronized through tools that control and track alternative versions (CVS), bug reports, or Web site content updates.

Accordingly, in order to explore where issues of collaboration, leadership, control and conflict may arise within or across related OSSD projects, then one place to look to see such issues is in how project participants create, update, exchange, debate, and make sense of the software informalisms that are employed to coordinate their development activities. This is the approach taken here in exploring the issues both within the NetBeans.org project community, as well as across the (fr)agile ecosystem [Highsmith 2002] of inter-related OSSD projects that situate NetBeans.org within a Web information infrastructure.

3. Intra-Community Issues

As noted in the first section, NetBeans.org is a large and complex OSSD project. To help convey a sense of the complexity, semi-structured modeling techniques such as rich pictures [Monk and Howard 1998] can be used to provide a visual overview of the context that situates the creation and manipulation of the software informalisms and OSSD processes that can be observed in the NetBeans.org project [Oza, *et al*, 2002]. Figure 1 displays such a rich picture, highlighting the variety of roles that participants in the NetBeans.org project perform, the types of concerns they have in each role, and the development tasks they regularly enact, as part of the configuration of OSSD activities they articulate and coordinate through software informalisms [cf . Simone and Mark 1999].

We have observed at least three kinds of issues arise *within* an OSSD community like NetBeans.org. These are collaboration, leadership and control, and conflict.

3.1. Collaboration

According to the Netbeans.org community Web site, interested individuals may participate in the community by joining in discussions on mailing lists, filing bug and enhancement reports, contributing Web content, source code, newsletter articles, and language translations. These activities can be done in isolation, without coordinating with other community members, and then offered up for consideration and inclusion. As we'll see, reducing the need for collaboration is a common practice in the community that gives rise to positive and negative effects. We discuss collaboration in terms of policies that support process structures that prevent conflict, looking at task completion guidelines and community architecture.

3.1.1. Policies and Guidelines

The NetBeans.org community has detailed procedural guidelines[1] for most common development tasks, from submitting bug fixes to user interface design and creating a new release. These guidelines come in two flavors: development task and design style guidelines. In general, these policies are practiced and followed without question. Ironically, the procedures for policy revision have not been specified.

Precedent states that revisions are brought up on the community or module discussion mailing lists, where they are debated and either ratified or rejected by consensus. Developers are expected to take notice of the decision and act accordingly, while the requisite guideline documents are updated to reflect the changes. In addition, as some communities resort to "public flogging" for failure to follow stated procedures, requests for revision are rare and usually well known among concerned parties, so no such flogging is done within Netbeans.org.

Overall, these policies allow individual developers to work independently within a process structure that enables collaboration by encouraging or reinforcing developers to work in ways that are expected by their fellow community members, as well as congruent with the community process.

3.1.2. Separation of Concerns: an Architectural Strategy for Collaborative Success

Software products are increasingly developing a modular, plug-in application program interface (API) architectural style in order to facilitate development of add-on components that extend system functionality. This strategy has been essential in an open source arena that carries freedom of extensibility as a basic privilege or, in some cases, the right of free speech or freedom of expression through contributed source code. But this separation of concerns strategy for code management also provides a degree of separation of concerns in developer management, and therefore, collaboration.

In concept, a module team can take the plug-in API specification and develop a modular extension for the system using any development process in complete isolation from the rest of the community. This ability is very attractive to third-party contributors in the Netbeans.org community who may be uninterested in becoming involved with the technical and socio-political issues of the community, or who are unwilling or unable to contribute their source code back to the community. Thus, this separation of concerns in the Netbeans.org design architecture engenders separation of concerns in the process architecture. Of course, this is limited by the extent that each module in the Netbeans.org community is dependent on other modules.

Last, volunteer community members have periodically observed difficulties collaborating with volunteer community members. For example, at one point a lack of responsiveness of the (primarily Sun employed) user interface team[2], whose influence spans the entire community, could be observed. This coordination breakdown led to the monumental failure of usability efforts for a period when usability was arguably the most-cited reason users chose competing tools over Netbeans.org. Thus, a collaboration failure gave rise to product failure. Only by overcoming collaboration issues was Netbeans.org able to deliver a satisfactory usability experience[3].

3.2. Leadership and Control

Ignoring internal Sun (and third party) enterprise structure, there are five observable layers of the Netbeans.org community hierarchy. Members may take on multiple roles some of which span several of these layers. At the bottom layer are users, followed by source contributors, module-level managers, project level release managers (i.e. IDE *or* platform), and finally, community level managers (i.e. IDE *and* platform) at the top-most layer. Interestingly, the "management" positions are simply limited to coordinating roles; they carry no other technical or managerial authority. The release manager, for example, has no authority to determine what will be included in and excluded from the release[4]. Nor does s/he have the authority to assign people to complete the tasks required to release the product. The same is true of module and community

[1] http://www.netbeans.org.org/community/guidelines/

[2] http://www.netbeans.org/servlets/ReadMsg?msgId=531512&listName=nbdiscuss

[3] http://www.javalobby.org/thread.jspa?forumID=61&threadID=9550#top

[4] http://www.netbeans.org.org/community/guidelines/process.html

managers. Instead, their role is to announce the tasks that need to be done and wait for volunteers to accept responsibility.

Accountability and expectations of responsibility are based solely on precedent and volunteerism rather than explicit assignment, leading to confusion of the role of parties contributing to development. Leadership is not asserted until a community member champions a cause and while volunteerism is expected, this expectation is not always obvious. The lack of a clear authority structure is both a cause of freedom and chaos in open source development. Though often seen as one of its strengths in comparison to closed source efforts, it can lead to process failure if no one steps forward to perform critical activities or if misidentified expectations cause dissent.

The difficulties in collaboration across organizations within the community occasionally brought up in the community mailing lists stem from the lack of a shared understanding leadership in the community. This manifests itself in two ways: a lack of transparency in the decision making process and decision making without community consent. While not new phenomenon, they are especially poignant in a movement whose basic tenets include freedom and knowledge sharing.

3.2.1. Transparency in the Decision Making Process

In communities with a corporately backed core development effort, there are often decisions made that create a community-wide impact that are made company meetings. However, these decisions may not be explicitly communicated to the rest of the community. Likewise private communication between parties that is not made available on the community Web space or to the forwarded to other members is also hidden. This lack of transparency in decision-making process makes it difficult for other community members to understand and comply with the changes taking place if they are not questioned or rejected. This effect surfaced in the Netbeans.org community recently following a discussion of modifying the release process [cf. Erenkrantz 2003][5].

Given the magnitude of contributions from the primary benefactor, other developers were unsure of the responsibility and authority Sun assumed within the development process. The lack of a clearly stated policy outlining these bounds led to a flurry of excitement when Sun members announced major changes to the licensing scheme used by the community without any warning. It has also caused occasional collaboration breakdown throughout the community due to expectations of who would carry out which development tasks. The otherwise implicit nature of Sun's contributions in relation to other organizations and individuals has been revealed primarily through precedent rather than assertion.

3.2.2. Consent in the Decision Making Process

Without an authority structure, all decisions in development are done through consensus, except among those lacking transparency. In the case of the licensing scheme change, some developers felt that Sun was within its rights as the major contributor and the most exposed to legal threat [6] while others saw it as an attack on the "democratic protection mechanisms" of the community that ensure fairness between participating parties[7]. A lack of consideration and transparency in the decision making process tend to alienate those who are not consulted and erode the sense of community.

3.3. Conflict Resolution

Conflicts in the Netbeans.org community are resolved via community discussion mailing lists. The process usually begins when one member announces dissatisfaction with an issue in development. Those who also feel concern with the particular issue then write responses to the charges raised. At some point, the conversation dissipates- usually when emotions are set aside and clarifications have been made that provide an understanding of the issue at hand. If the problem persists, the community governance board is tasked with the responsibility of resolving the matter.

The governance board is composed of three individuals and has the role of ensuring the fairness throughout the community by solving persistent disputes. Two of the members are elected by the community, and one is appointed by Sun Microsystems. The board is, historically, a largely superficial entity whose authority and scope are questionable and untested. While it has been suggested that the board intercede on a few rare

[5] http://www.netbeans.org/servlets/BrowseList?listName=nbdiscuss&by=thread&from=19116&to=19116&first=1&count=41

[6] http://www.netbeans.org/servlets/ReadMsg?msgId=534707&listName=nbdiscuss

[7] http://www.netbeans.org/servlets/ReadMsg?msgId=534520&listName=nbdiscuss

occasions, the disputes have dissolved before the board has acted. Nevertheless, board elections are dutifully held every six months[8].

Board members are typically prominent members in the community. Their status carries somewhat more weight in community policy discussions, however, even when one member has suggested a decision, as no three board members have ever voted in resolution on any issue, and thus, it is unclear what effect would result. Their role, then, is more of a mediator: to drive community members to resolve the issue amongst themselves. To this end, they have been effective.

4. Inter-Community Issues

As noted earlier, the NetBeans.org project is not an isolated OSSD project. Instead, the NetBeans IDE which is the focus of development activities in the NetBeans.org project community is envisioned to support the interactive development of Web-compatible software applications or services that can be accessed, executed, or served through other OSS systems like the Mozilla Web browser and Apache Web server. Thus, it is reasonable to explore how the NetBeans.org project community is situated within an ecosystem of inter-related OSSD projects that facilitate or constrain the intended usage of the NetBeans IDE. Figure 2 provides a rendering of some of the more visible OSSD projects that surround and embed the NetBeans.org within a Web information infrastructure. This rendering also suggests that issues of like collaboration and conflict can arise at the boundaries between projects, and thus these issues constitute relations that can emerge between project communities in OSSD ecosystem.

With such a framing in mind, we have observed at least three kinds of issues arise *across* OSSD communities that surround the NetBeans.org community. These are communication and collaboration, leadership and control, and conflict resolution.

4.1. Communication and Collaboration

In addition to their IDE, Netbeans.org also releases a general application development platform on which the IDE is based. Other organizations, such as BioBeans and RefactorIT communities build tools on top of or extending the NetBeans platform or IDE. How do these organizations interact with Netbeans.org, and how does Netbeans.org interact with other IDE and platform producing organizations? For some organizations, this collaboration may occur in terms of bug reports and feature requests submitted to the Netbeans.org issue-tracking repository. Additionally, they may also submit patches or participate in discussions on community mailing list or participate in the Netbeans.org "Move the Needle" branding initiative. Beyond this, Netbeans.org participates in the Sun sponsored *Java.net* meta-community, which hosts hundreds of Java-based OSSD projects developed by tens of thousands of individuals and organizations.

A fellow member of the Java.net community, the Java Tools Community, considered by some to be a working group[9] for the Java Community Process, is an attempt to bring tool developers together to form standards for tool interoperability. Thus Netbeans.org, through its relationship with Sun, is a collaborating community in the development of, and through compliance with, these standards, and looks to increasing collaboration with other tool developing organizations.

4.2. Leadership and Control

OSSD generally embrace the notion of choice between software products to build or use. At the same time, developers in any community seek success for their community, which translates to market share.

In some cases, communities developing alternative tools do so in peaceful coexistence, even collaboratively. In other cases, there is a greater sense of competition between rivals. NetBeans and its chief competitor Eclipse (backed largely by IBM) fall into the latter category. Eclipse has enjoyed some favor from users due to performance and usability issues of NetBeans, as well as IBM's significant marketing and development resource contributions. Yet, they have a willingness to consider collaborative efforts to satisfy demands for a single, unified IDE for the Java language that would serve as a platform for building Java development tools and a formidable competitor to Microsoft's .NET. Ultimately, the union was defeated, largely due to technical and organizational differences between Sun and IBM[10], including the inability or

[8] http://www.netbeans.org.org/about/os/who-board.html

[9] http://www.internetnews.com/dev-news/article.php/3295991

[10] http://www.adtmag.com/article.asp?id=8634, and

unwillingness to determine how to integrate the architectures and code bases for their respective user interface development frameworks (Swing for NetBeans and SWT for Eclipse).

4.3. Conflict Resolution

Conflicts between collaborating communities are resolved in similar fashion to their means of communication- through discussion between Sun and Eclipse representatives, comments on the Netbeans.org mailing lists, or other prominent technical forums (e.g. Slashdot and developer blogs). Unfortunately, many of these discussions occur after the collaborating developer has moved away from using Netbeans.org (often, in favor of Eclipse). Nevertheless, the feedback they provide gives both parties an opportunity to increase understanding and assists the Netbeans.org community by guiding their technical direction.

5. Discussion and Conclusion

Generally, volunteer Netbeans.org developers expect Sun to provide leadership but not control. People outside the community (e.g. users, former users, and potential users) often voice their concerns in off-community forums (e.g., Slashdot, blogs, etc) rather than NetBeans.org community message boards, due to accountability or visibility barriers (creating an account, logging in accounts), small as they may seem to be. In addition, such message forums may not be a part of such an individual's daily work habits- they're more likely to visit a site like Slashdot.org than the Netbeans.org forum because they are not interested enough in staying abreast of NetBeans developments or participating in the community. Nonetheless, people working in, or interested in joining or studying OSSD projects, must address how best to communicate and collaborate their development processes and effort, how to facilitate or ignore project leadership and control, and how to work you way through conflicts that may or may not be resolvable by community participants.

Overall, we have observed three kinds of coordination and collaborating issues arise within OSSD project communities like NetBeans.org, and three similar kinds of issues arise across OSSD communities that surround NetBeans.org within an ecosystem of projects that constitute a Web information infrastructure. Previous studies of conflict in either OSSD projects have examined either smaller projects, or in virtual communities that do not per se develop software as their focus. As corporate interest and sponsorship of OSSD stimulates the formation of large projects, or else the consolidation of many smaller OSSD projects into some sort of for-profit or not-for-profit corporate enterprise for large-scale OSSD, then we will need to better understand issues of collaboration, conflict, and control in OSSD.

6. Acknowledgments

The research described in this report is supported by grants from the National Science Foundation #ITR-0083075 and #ITR-0205679 and #ITR-0205724. No endorsement implied. Contributors to work described in this paper include Mark Ackerman at the University of Michigan Ann Arbor; Les Gasser at the University of Illinois, Urbana-Champaign; John Noll at Santa Clara University; and Margaret Elliott at the UCI Institute for Software Research.

7. References

Augustin, L., Bressler, D., and Smith, G., Accelerating Software Development through Collaboration, *Proc. 24th Intern. Conf. Software Engineering*, Orlando, FL, 559-563, May 2002,

Crowston, K. and Scozzi, B., Open Source Software Projects as Virtual Organizations: Competency Rallying for Software Development, *IEE Proceedings—Software*, 149(1), 3017, 2002.

Easterbrook, S. (ed.), *CSCW: Cooperation or Conflict*, Springer-Verlag, New York, 1993.

Elliott, M. The Virtual Organizational Culture of a Free Software Development Community, *Proc. 3rd Workshop on Open Source Software Engineering, 25th. Intern. Conf. Software Engineering*, Portland, OR, May 2003.

Elliott, M. and Scacchi, W., Free Software Developers as an Occupational Community: Resolving Conflicts and Fostering Collaboration, *Proc. ACM Intern. Conf. Supporting Group Work*, 21-30, Sanibel Island, FL, November 2003.

Erenkrantz, J., Release Management within Open Source Projects, *Proc. 3rd Workshop on Open Source Software Engineering, 25th Intern. Conf. Software Engineering*, Portland, OR, May 2003.

Highsmith, J., *Agile Software Development Ecosystems,* Addison-Wesley Pub. Co., 2002.

http://www.eweek.com/article2/0,1759,1460110,00.asp

Jensen, C. and Scacchi, W., Automating the Discovery and Modeling of Open Source Software Processes, *Proc. 3rd Workshop on Open Source Software Engineering*, 25th. Intern. Conf. Software Engineering, Portland, OR, May 2003.

Jensen, C. and Scacchi, W., Process Modeling of the Web Internet Infrastructure, submitted for publication, June 2004.

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

O'Mahony, S., Non-profit Foundations and their Role in Community-Firm Software Collaboration, to appear in *Making Sense of the Bazaar: Perspectives on Open Source and Free Softwar*e, J. Feller, B. Fitzgerald, S. Hissam, & K. Lakhani (Eds.), O'Reilly & Associates, Sebastopol, CA, 2004.

Sawyer, S., Effects of intra-group conflict on packaged software development team performance, *Information Systems J.,* 11, 155-178, 2001.

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

Scacchi, W., Free/Open Source Software Development Practices in the Computer Game Community, *IEEE Software*, 21(1), 59-67, January/February 2004.

Schmidt, K., and Simone, C., Coordination Mechnanisms: Towards a Conceptual Foundation of CSCW System Design, *Computer Supported Cooperative Work,* 5(2-3), 155-200, 1996.

Simone, C. and Mark, G., Interoperability as a Means of Articulation Work, *Proc. Intern. Joint Conf. Work Activities Coordination and Collaboration,* San Francisco, CA, 39-48, ACM Press, 1999.

Smith, A.D., Problems of Conflict Management in Virtual Communities. In M.A. Smith and P. Kollock (eds.), *Communities in Cyberspace,* Routledge, New York, 134-163, 1999.

Star, S. L., The Structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous Distributed Problem Solving, in *Distributed Artificial Intelligence* (eds. L. Gasser and M. N. Huhns), Vol. 2, 37-54. Pitman, London, 1990.

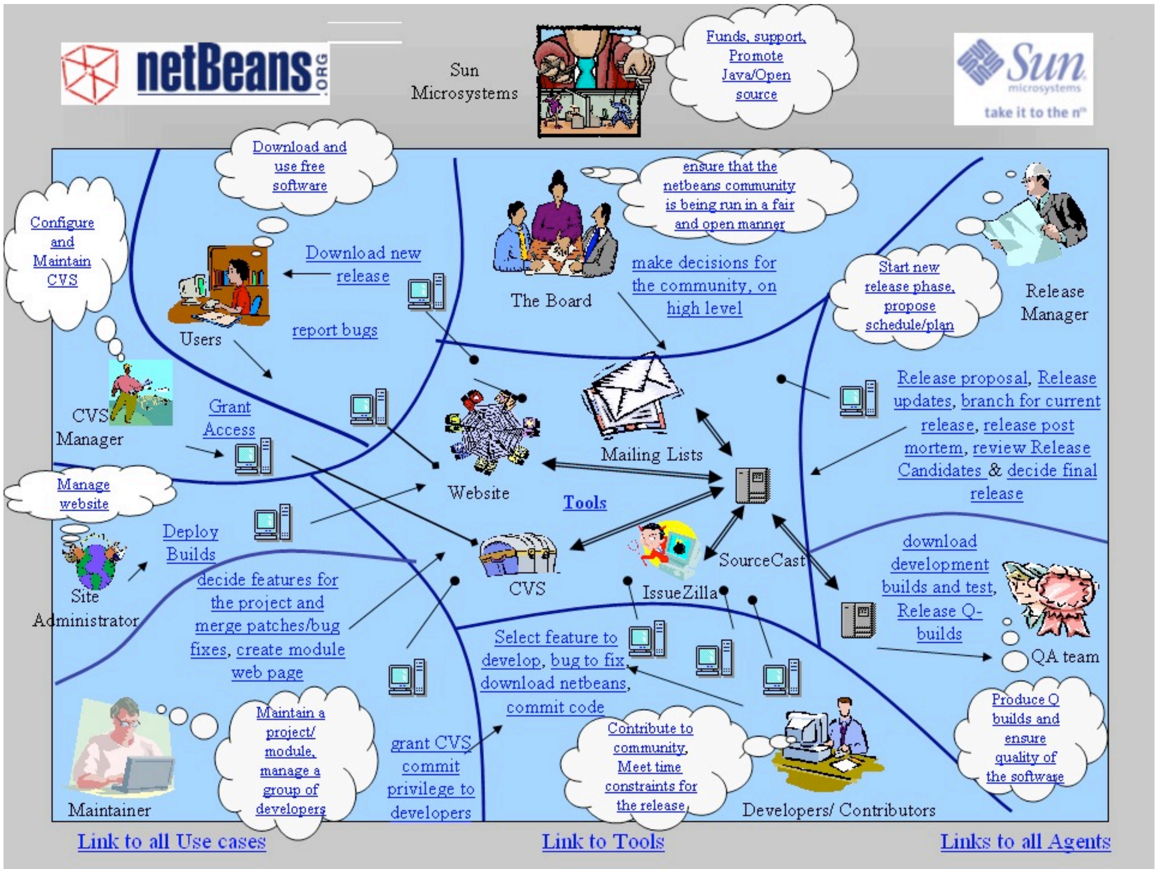
netBeans.ORG
Sun Microsystems
Funds, support, Promote Java/Open source
Sun microsystems take it to the nth
Download and use free software
Configure and Maintain CVS
Download new release
Users
report bugs
The Board
ensure that the netbeans community is being run in a fair and open manner
make decisions for the community, on high level
Start new release phase, propose schedule/plan
Release Manager
CVS Manager
Grant Access
Website
Mailing Lists
Release proposal, Release updates, branch for current release, release post mortem, review Release Candidates & decide final release
Manage website
Tools
Deploy Builds
SourceCast
Site Administrator
decide features for the project and merge patches/bug fixes, create module web page
CVS
IssueZilla
download development builds and test, Release Q-builds
QA team
Select feature to develop, bug to fix, download netbeans, commit code
Maintain a project/ module, manage a group of developers
grant CVS commit privilege to developers
Contribute to community, Meet time constraints for the release
Produce Q builds and ensure quality of the software
Maintainer
Developers/ Contributors
Link to all Use cases
Link to Tools
Links to all Agents

Figure 1. A rich picture view of the roles (labeled icons), concerns (clouds), and activities (hyperlinked text) found in the intra-community context of NetBeans.org [cf. Oza, *et al*, 2002].

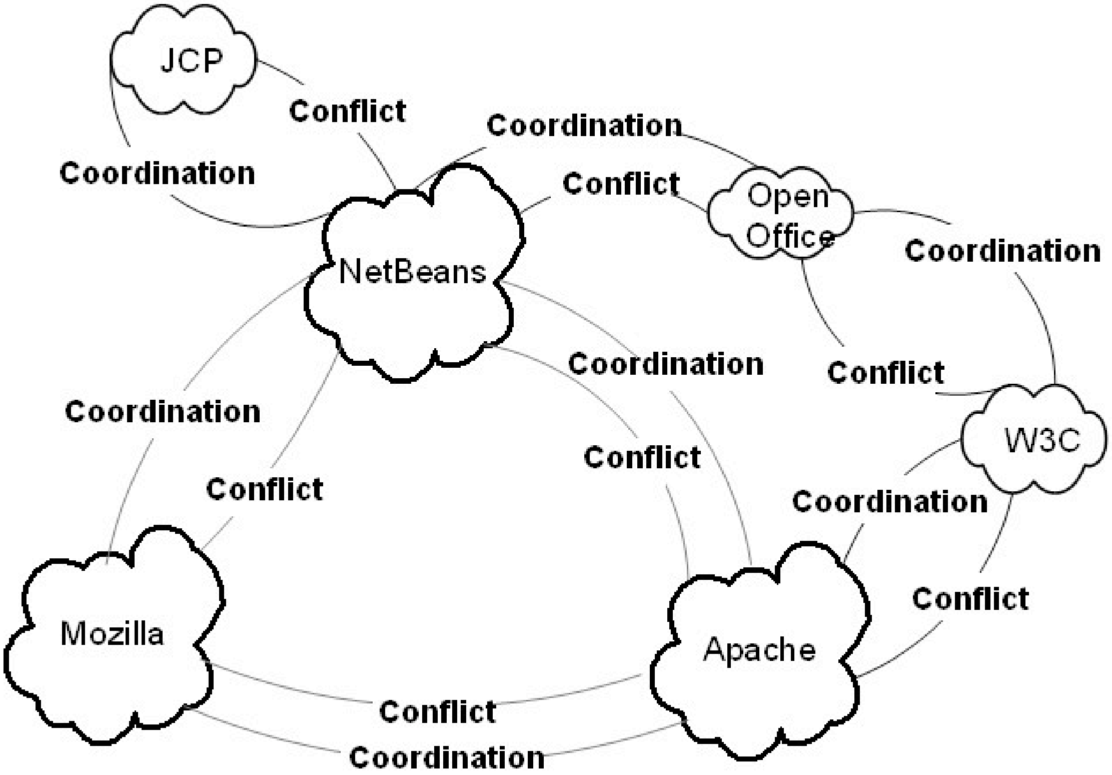

Figure 2. An overview of the inter-community ecosystem for NetBeans.org