

Communicating and Mitigating Conflict in Open Source Software Development Projects

Margaret S. Elliott

Institute for Software Research
University of California, Irvine
Irvine, CA 92697
949 824-4310
melliott@ics.uci.edu

Walt Scacchi

Institute for Software Research
University of California, Irvine
Irvine, CA 92697
949 824-4130
wscacchi@ics.uci.edu

ABSTRACT

In this paper, we present results from a study of how conflict emerges, is communicated, mitigated and resolved in a globally dispersed open source software development project. In the study, conflict appears in development of an open source business office system (BOS). The BOS open source developers are characterized as an occupational subculture. These developers resolve their conflict through interactions communicated via Internet-Relay Chat, persistent chat logs, threaded email discussions, and community digests. We show how cultural beliefs in "free software" are manifested in software development methods, artifacts, and tool choice, as well as how dispersed developers cooperate.

Keywords

Computer-supported cooperative work, conflict, CSCW, occupational culture, open source software development

INTRODUCTION

Computer-supported cooperative work (CSCW) involves both cooperation and conflict (Easterbrook, 1993). Conflict is an inherent part of the culture of work since "conflict and cooperation are omnipresent and inevitably coexistent in social life (Watson, 1987, p. 213)." Researchers recognize the importance of conflict in CSCW and suggest that designers of CSCW systems need to pay more attention to conflict and its management when building CSCW systems (Easterbrook, 1993). However, more research is needed to better understand how conflicts are resolved and managed in cooperative system development work environments, particularly those involving people who build software at a distance, and rarely if ever meet for face-to-face interaction.

In this paper, we present a qualitative study of conflict and open source software development (OSSD) in a virtual community where persistent Internet Relay Chat (IRC) logs (i.e., transcripts) and threaded email discussions provide a persistent record of work practices and serve as a resource embodying organizational memory. These IRC logs are reminiscent of the online discourses of conflict

and cooperation articulated, and sometimes captured and made persistent, in MUDs (Muramatsu and Ackerman 1998). We show how these IRC logs, as "lean media" (Yamaguchi, *et al.*, 2000), serve as an aid to conflict resolution and how occupational subculture beliefs and values influence the OSSD process. The posting of the chat logs, email discussions, and summary digests on a public Web site embody the spirit of the free software movement and serve as a point of entry, exposure, and enculturation for both new and experienced open source software developers.

Researchers have studied the effects of conflict on closed source software development efforts, but not in the OSSD efforts. For example, Sawyer (2001) surveyed 40 packaged software development teams and concluded that how people work together and mitigate conflict is more important than individual skills and abilities of the team's members in predicting their software development performance. Furthermore, the developers Sawyer studied developed commercial, closed source software packages, and thus were likely able to meet in face-to-face project meetings, within centralized and hierarchically managed corporate settings. Elsewhere, Carmel (1999) provides an analysis of globally dispersed software development projects, highlighting the role that differences in national culture make in inhibiting or facilitating technical discussions and decision-making. But Carmel also focuses on corporate settings developing closed source software systems.

Thousands of OSSD projects have emerged within the past few years (see <http://www.sourceforge.net>) leading to the formation of globally dispersed communities of practice (Wenger 1998). Examples of open source projects are found in the social worlds that surround computer game development; X-ray astronomy and deep space imaging; academic software design research; and Internet/Web infrastructure development (Scacchi 2001). In communities such as these, OSS developers work as peers who rely on Web-based computing environments to support and coordinate their development work in globally decentralized settings.

Proponents of OSSD hail advantages such as improved software validity, simplification of collaboration, and reduced software acquisition costs (DiBona, *et al.*, 1999, Pavlicek 2000). However, few empirical studies have been conducted to validate or explore claims like these (e.g., Koch and Schneider 2002, Yamamoto *et al.*, 2000, Sharma, *et al.*, 2001). Research has focused on the quantitative side of OSSD projects, such as aspects of developer defect density and core team size (e.g., Mockus *et al.*, 2000, Stamelos, *et al.*, 2002). More studies are needed using a socio-technical perspective to develop empirically grounded understandings of the social circumstances surrounding the technical system configurations and virtual organizational contexts that comprise OSSD projects.

We present the results of a study of an OSSD project here called Business Office Systems (BOS), where conflict is resolved through interactions communicated through collaborative, lean media like IRC and email forums. Occupational subcultures (Trice and Beyer, 1992; Van Maanen and Barley, 1984) are used to frame our interpretive analysis. Researchers have addressed cultural dimensions of computing (Star, 1996) as an important aspect of social analyses of computing. For example, Baym (1996) completed a cultural analysis of an online discussion group whose purpose was to exchange comments on television soap operas. She used the practice approach to examine the cultural framing of individual messages posted in an email discussion forum. In her analyses, community culture can be viewed as continually changing through the actions and expressions of the participants making practical choices. For more examples and a detailed discussion of cultural studies of computing in organizations, see Elliott (2000).

In this study, we use occupational subcultures as an alternative lens to bring into focus the cultural beliefs and values that contextualize the collaborative communication and mitigation of conflicts in loosely coordinated software development work. We use a comparative three perspective view, as developed by Martin (1992), to examine the occupational subculture that helps situate the BOS development effort. Dube and Robey (2000) also use these three perspectives to interpret data from their study of a software development company's management practices - development team organization and outsourcing. Their results from this multi-perspective analysis of stories told to them by software developers, point to the importance of understanding the cultural foundation of management practices used in software development.

We apply the three-perspective cultural view to the BOS project, an OSSD effort where there are no corporate guidelines, central administrative authorities, or management cues to control and coordinate software development. There is instead a loosely knit community of

developers and others who use a networked, computer-supported work environment primarily constituted by Web-based repositories, Web browsers, messaging systems, and local software development tools as their virtually collocated workplace and workspace. In these settings, no one has the power to hand down decisions from above (Pavlicek 2000). Other significant characteristics of the BOS setting from a CSCW standpoint include:

- * Routine public recording, posting, browsing, and referencing of online project meeting transcripts.

- * No cues are available from face-to-face meetings, so text messaging is used for social cues. Participants are challenged to understand each other.

- * English is a second language for some participants, so dimensions of national culture are part of the interaction and communication process¹.

- * Global participation results in either temporal collocation or dislocation.

- * Individual developer identities in the real world are known, in comparison with other cyber-communities where people remain anonymous.

- * Volunteers do the OSSD work, though some are paid by outside firms to work on the BOS, some have other jobs, and still others not paid.

We show how three recurring cultural beliefs and values of this community - belief in free software practices and principles, belief in free speech about software development processes, and freedom of choice in selecting work - directly influence the communication and mitigation of conflicts in software development practices.

In the next section, we present the background on occupational subcultures followed by a description of the three perspectives on organizational culture. Next we present the research methodology, the BOS description, and BOS content themes. Finally, we end with a comparative analysis of our data and our conclusions.

OCCUPATIONAL SUBCULTURES

One of the most widely studied and distinctive sources of organizational subcultures is based on people's occupations: *occupational subcultures* (Trice and Beyer 1993, Schein 1997, Van Maanen and Barley 1984). Two different types of subcultures form from occupational cultures: 1) diffuse subcultures in general society, and 2) face-to-face subcultures within organizations. In this study of BOS, the OSS developers are considered an

¹ Participants also must comprehend and act through emoticons ☺, chat conventions (e.g., "lol" for laugh out loud), idioms, and various kinds of humor (ironic cliches, sarcasm, etc.), all across diverse national cultures.

occupational subculture within the overall occupational culture of software developers. As Gregory (1983) has shown, software developers working in Silicon Valley shared the same occupational subculture, no matter which organization they worked in, or what kind of work they did.

Researchers have identified characteristics of occupational subcultures (Trice and Beyer, 1993). The list presented below identifies those that apply to OSS developers as an occupational subculture:

Reference groups - Members develop shared ideologies by using one another as reference points. They react to other members behavior and inculcate beliefs, values, and norms from each other into their view of themselves in the occupational community. (Open source developers use each other as reference points in their ideologies about open source).

Favorable self-image and social identity - members derive favorable self-images and social identifies from their work that is then projected to others. (Open source culture fits this category).

Extension into non-work life - Members socialize outside of their physical workplace. Off project chats allow open source developers to talk about themselves and matters in their personal lives, in the course of their daily work. (Online chats become social events).

THREE PERSPECTIVES ON ORGANIZATIONAL CULTURE

Martin (1992) presents three perspectives on organizational culture to articulate a more complete picture of an organization via comparative analyses. Martin perspectives are:

* Integration - Culture is defined as that which is shared by a given organization. Emphasis is on leadership-oriented cultural change and/or on consistency and consensus among cultural members.

* Differentiation - Culture is viewed as resulting in inconsistencies, lack of consensus and non-leader centered sources of cultural content.

* Fragmentation - Culture is viewed as having no shared values except one: the awareness of ambiguity.

We use each of these perspectives to analyze how occupational subculture influences choices made about what tools/systems to use to support development, how development artifacts should be created, and how cultural beliefs and values surrounding OSSD are expressed and reproduced, in the BOS project.

RESEARCH METHODOLOGY

The data presented here is part of an ongoing study using the grounded theory approach (Glaser and Strauss, 1967). The sources of data include books and articles on OSS development, instant messaging (Herbsleb, *et al.*, 2000, Nardi, *et al.* 2000) transcripts captured through IRC logs,

threaded email discussion messages, and other Web-based artifacts associated with BOS. The data we present below is publicly available on the Web. However, our ethical choice is to mask the identity of the project and participants in our study with pseudonyms, so as to not draw undue attention nor encourage judgement of the project, its participants, their beliefs, or their actions.

BUSINESS OFFICE SYSTEMS (BOS)

Business Office Systems (BOS) is an OSSD project whose objective is a complete enterprise-level business environment. BOS developers communicate from around the world via email, mailing lists, active IRC sessions, passive IRC logs, and Web site documents to coordinate who is working on what, what version gets released, and other work practices. BOS itself consists of tools to build business applications, from either ready-made modules, or from easily modified templates to suit individual business needs. Designed in a modular fashion, BOS objects are defined at application run-time enabling frequent reconfigurations as needed. The BOS software is developed for use with the GNU/Linux operating system using free software tools. There are five core contributors dispersed across the U.S. and Europe, and 21 additional contributors from Belgium, Canada, England, Estonia, France, Italy, New Zealand, Spain, and the U.S, according to developer identity information on the Web site. Most of these people may never meet face-to-face, yet they work together as a virtual team to develop BOS. One effective way of communicating and sustaining what is going on in the BOS project is through a summary digest of project email discussions, IRC logs, as well as other hyperlinked artifacts hosted on the project's Web site. A BOS contributor, often a project observer rather than an OSS developer, volunteers to create and maintain such a digest in a common format. Each digest is cleverly indexed by subject headings selected from the email records and IRC logs. The summaries read more like stories (Dube and Robey 1999) than dry iterations of the archived content. However, these digests provide an entry way into the BOS project for new developers and observers who may want to join the ongoing effort.

RECURRING BELIEFS AND VALUES IN BOS

In this section, we present three beliefs derived from an analysis of the data. Though we find many kinds of beliefs in the data, we selected three recurring expressions as representative of belief in the precepts of building free/open source systems: (1) belief in free software practices and principles; (2) belief in open disclosure of OSSD methods of software development; and (3) belief in freedom of choice in work assignments. Each of these is discussed in relation to how they are manifested in formal and informal work practices, and in work artifacts. It is beyond the scope of this paper to present other manifestations of the BOS culture such as symbols, rituals, or Web site design (Rivett 2000).

Belief in Free Software Practices and Principles

The BOS developers show a strong belief in free software extolling its virtues on its Web site and in daily activity on the IRC logs. This belief is manifested in electronic artifacts such as the Web pages, source code, software design diagrams, and accompanying articles. The BOS Web site advertises that it is "a free software project with a corps of volunteer developers around the world working on BOS projects". The Web site provides a link to an article by Richard Stallman, developer of the free/open source Emacs text editor and GNU C compiler, as well as a founder and leading advocate of the Free Software Foundation. Stallman is well recognized for extolling the philosophy of free software. In this article at <http://www.gnu.org/philosophy/why-free.html>, he states:

"As a computer user today, you may find yourself using a proprietary program. If your friend asks to make a copy, it would be wrong to refuse. *Cooperation is more important than copyright.* But underground, closet cooperation does not make for a good society. A person should aspire to live an upright life openly with pride, and this means saying 'No' to proprietary software. *You deserve to be able to cooperate openly and freely with other people who use software.* You deserve to be able to learn how the software works, and to teach your students with it. You deserve to be able to hire your favorite programmer to fix it when it breaks. You deserve free software." (Emphasis added)

The BOS project is advertised as free software for business applications on its Web site. The BOS software is licensed under the GNU General Public License (GPL) (<http://www.gnu.org/copyleft/gpl.html>). The preamble to this license states the philosophy behind the free software approach:

"The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to *guarantee your freedom to share and change free software*--to make sure the software is free for all its users ... When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and *that you know you can do these things.*" (Emphasis added).

The belief in free software is manifested formally, through the rights and imperatives afforded in the GPL that one realizes if employing free software, and informally in the moral imperatives (emphasized in the quoted excerpts) that contextualize the software development work

practices of OSSD contributors. Throughout the digests and IRC logs, there are numerous references to the importance of adhering to the principles of free software. Often arguments are recorded on the BOS IRC logs related to using non-free software to produce free documentation and code.

Five core contributors, those people who take responsibility for specific areas of BOS development, are "regulars" on the daily IRC logs. Some of these contributors are paid by outside firms to work on BOS, while others are unpaid for their contribution to BOS development. In addition, there are other frequent and infrequent contributors to these logs. Some people log on and never contribute conversation, but "listen" or lurk for hours at a time, as "participant observers."

We now present excerpts from a log focused on a conflict trajectory, from emergence through debate and mitigation. This conflict arose when an infrequent contributor investigates the origin of a BOS system design graphic posted on the Web site. We took the complete eleven page (single space) IRC log and divided it up into segments that illustrate the communication and mitigation of this conflict (cf. Marumatsu and Ackerman 1997). Extraneous comments have been omitted for presentation here, while the IRC dialogue entries are numbered for clarity. The lines beginning with "Action:" represent a signal asserted by a participant that s/he is doing something concurrent to participating in the active IRC.

In this initial passage, a newcomer to the BOS IRC, Jim raises the issue of using non-free software for graphic documentation of BOS on the Web site, and BobR, a core contributor, responds:

1. <Jim> Several images on the BOS website seems to be made with non-free Adobe softwares, I hope I'm wrong: it is quite shocking. Does anybody know more on the subject? We should avoid using non-free software at all cost, am I wrong?

2. <BobR> Jim: Our main goal is to produce good free software. We accept contributions without regarding what tools were used to do the work. Especially we accept documentation in nearly any form we can get because we are desperate for documentation... just as long as the format itself isn't proprietary, and it can be viewed without proprietary programs anything is ok for us. But if you want to redo those pictures in dia (or whatever) we will gladly take it. The contributor was not familiar with dia at all and felt that he would be more productive when he used his adobe... and we were ok with that.

3. <Jim> I understand your point of view but if you accept contributions that can be viewed with free software, you also have to be able to modify the contributions. What if we need to add a component to this

graphic? Even ASCII art graphic would be better.

4. <BobR> Jim: isn't png viewable with free software?

5. <Jim> BobR: png is viewable with free software, you are right. You can consider this PNG as a binary distribution of the contribution, not the source code

Lines 1-5 show how belief in free software drives Jim to react to non-free graphics. Despite Jim's relative newness to the BOS community, his views are respected and participants begin a lengthy discussion on the merits of free versus non-free software in development. Do they allow the graphic made with non-free software to remain or do they require the developer to replace it with one made from free software? BobR suggests that using the Adobe non-free software was acceptable to the BOS community. Jim is not satisfied and pursues his preference for use of free software with determination:

6. <Jim> We NEED to be able to modify the code. And we can't modify adobe files with free softwares.

7. <BobR> so we need someone that can do the graphic in dia?

8. <Jim> We need people do be able to use free softwares and produce free documents.

9. <BobR> You know someone who would want to do this graphic as well as maintain it for adding new modules etc over the next few years? i think we have to seek such a person till we found it i think we can live with what we have now as an intermediate solution

10. <Jim> If this solution is considered as an intermediate solution, it is ok for me.

In the next passage, Jim establishes his identity as an affiliate of the European Free Software Foundation, and as a first time participant in a BOS IRC-based project meeting. Jim is from France and despite his newness to the BOS community, Bob and Nathan take his suggestion and criticism seriously. They bond together for the purpose of developing free software and try to resolve the problem:

11. <BobR> any other comments on BOS? iirc i saw you a few times in #fsfeurope but never here on BOS?

12. <Jim> This is the first time for me in BOS. Did the author of this graphic understood that this file has to be freed? I think that if he is able to produce this kind of graphics with non-free softwares, he can easily do the same with free softwares.

13. <BobR> From what i know about the author i think he is aware of the issue but he works on mac mostly and afaik not so much free software is available for mac

14. <Jim> This discussion is interesting and I have to talk much with you later but I have to go outside now.

15. Nathan joined #BOS.

16. <BobR> hello Nathan. i just had a discussion about the graphics you did for the homepage. i hope what i said is ok for you

17. <Nathan> hello

18. Action: Nathan goes to look at log

19. <Nathan> BobR: it would be nice of we had free software that would do nice diagrams. it does not exist

20. <BobR> IIRC i have never made such a diagram in my entire life neither with proprietary nor with free software :) so i can't tell but i can imagine very well you're right :)

21. <Nathan> you do know that my graphics are the most viewed screenshots on the web site and it was not me sitting here hitting the reload button :)

22. <BobR> lol

Nathan reviews the log but does not understand Jim's objections to the graphic. Next Nathan and BobR discuss the technical details of whether or not his diagram is in pixel or vector format:

23. <Nathan> i think i missed the point. What is the problem with the graphics?

24. <BobR> His point was that only you have the "source" and only you can change the graphic. if i understand him correctly, as you are probably the only one among us that has this adobe software.

25. <Nathan> Any program that reads png which is a standard format can edit the graphics. if you can suggest another format, Kevin said this was the preferred for graphics, then I will convert to a better format

26. <BobR> i think his point is that usually you create such a graphic with a vector drawing tool. However this is somewhat "silent post"

27. <Nathan> png is a vector format graphic, so all vector information is in the version on the web

28. <BobR> i thought png is a pixel graphic format, but i can be wrong i'm not sure at all

29. <Nathan> portable network graphic

For the next 27 lines of the IRC log, BobR views the vector information stored with the graphic and convinces Nathan that png is a pixel format. Next Nathan asks for help from participants on the IRC. The non-participating lurkers offer suggestions on various "free" graphic programs that he could use. Jim returns:

30. <BobR> ...i think he[Nathan] got your point

31.<Jim> The point is not vector drawing or pixel drawing, as you say I don't deal with non ascii files. The point is free software VS non-free software.

32.<BobR> oh

33.<Jim> Nathan: welcome back.

34.<Jim> <Nathan> BobR: it would be nice of we had free software that would do nice diagrams (...) it does not exist

35. <Nathan> Jim: what do you suggest

Jim suggests free software - dia or xfig - for graphics and speaks disparagingly about Nathan's Adobe software. Finally, they argue again about the merits of free versus non-free software with BobR joining to support Jim:

36.<Jim> Nathan: friends of mine are using dia and xfig for this kind of graphics.

37.<Nathan> what is xfig

38.<BobR> dia can't even display the same font on screen as it does when printing out

39.<Jim> Nathan: dia [is not as bad as] adobe non-free softwares but dia [is bad], you are right

40.<Nathan> Jim: i am sorry, but i am not a biggot about software

41.<Jim>

<http://freshmeat.net/projects/xfig/>

42.<Nathan> my time is valuable. anything that wastes my time is not good, free or not

43.<Jim> What is 'biggot'?

44.<Nathan> biased

45.<Jim> Nathan: I don't agree at all, we should use free software at all costs.

46.<BobR> Jim: no i don't agree here. We should develop good free software at all costs

47.<Jim> BobR: using non-free software _do not_ 'develop good free software'.

48.<BobR> it can

49.<Jim> and promoting them is really a shame.

50. <Jim> BobR: not in this case.

Despite the controversy that Jim has caused, it is surprising that Nathan is receptive to changing his graphic and even installs xfig during the IRC:

51.Action: Nathan does install xfig

52.<Nathan> installing now

53.<BobR> i don't see that this is promoting. a "normal user" doesn't see where this png comes from.

54.<Jim> BobR: Nathan just said Adobe non-free software make him avoid loosing time.and free software DO make him loose time

55.<Nathan> Jim: i agree to goal BOS, that is to use free software for stuff in cvs

56.Action: Jim is shocked

57.<Nathan> so if there is a free software alternative, i will support that

58.<Jim> Nathan: I've used xfig a couple of time, I can help you.

59.<Nathan> i did not know xfig existed. i am installing now. i'll let you know how it works. otoh i see no reason to avoid non-free software either if this is really a freedom thing then we should be free to use whatever we want.

60.<BobR> Nathan: i agree, as long as we don't take away freedom from others. For example, i think it's ok if i use vi, emacs or even windows notepad to write my source code but if i used winword and stored as .doc i would take away freedom from those wanting to read the text

61. <Nathan> in your case you are saying its not about freedom i guess, its about using what the free software movement tells you to use that is just another form of bias. Again only because i am being told that using non-free software is bad :(i wish we could just leave the opinions about free vs non-free out

62. <BobR> the main point is that we want to achieve something a very ambitious goal actually in this project and everyone tries is best to reach that goal his best and we make compromises if it helps the overall goal anyway gotta get me some sleep

Jim and Nathan continue the discussion about xfig versus dia. Nathan becomes testy and asks Jim what he does professionally when arguing about changing the graphic:

63.<Nathan> Jim: ok i installed xfig. it does not show up in gnome. Any ideas where it went?

64.<Jim> Nathan: what do you mean by 'not show up in gnome' ? Does it has to 'appear' magically somewhere ?

65.<Nathan> yes, i found it. it looks like my old mac 128 circa 1986

66.<Jim> Nathan: the interface is ugly but useful.

67. <Nathan> Jim: sorry but i have to suffer through this, because you think its better why? What do you do professionally?

Throughout the interchange between Nathan and Jim, Nathan is also exchanging information with several contributors concerning the technical merits of dia and xfig. Meanwhile, Jim continues to insist on free software for the graphic. Finally, Nathan wants to know Jim's credentials:

68.<Jim> Nathan: you are compromising our freedom by using non-free software: we can't modify and/or redistribute the source vector file.

69. <Nathan> so we should change to *.xfig because its such a popular format?

70. <Jim> Nathan: you don't need colors
71. <Nathan> Jim: thats pure crap
72. <Jim> Nathan: not popular but FREE
73. <Nathan> Jim: what do you do professionally?
74. <Jim> Nathan: sysadmin in a european isp
75. <Nathan> Jim: cool, thanks

Finally, Jim makes his exit apologizing to Nathan for his criticism and they part on amicable terms:

76. <Jim> I have to leave you now.
77. <Nathan> Jim: later
78. <Jim> Nathan: excuse me if I said stupid things.
79. <Nathan> Jim: no problem, i dont takes things badly
80. <Jim> Nathan: do not hesitate to contact me if you have any questions with xfig: [email address]
81. <Nathan> i just feel free to express myself as everyone else should.
82. <Nathan>Jim: thank you

During this discussion, notice in line 67 and again in line 73, Nathan asks Jim what he does professionally. Jim responds with "Sysadmin in European isp." He revealed earlier to BobR that he is from the European Free Software Foundation, and Jim reads this in the log. It appears that these occupational credentials give Jim's comments more credibility for Nathan. Eventually, Nathan and Jim come to some agreement for the graphic to be redone using free software and part on better terms.

The tension between free versus non-free software results in conflict that is then mitigated through communication on the active IRC channel, and by cooperative work between contributors. These people work in a virtual world and use this as a medium to solve problems. The belief in free software is a driving force behind the cooperative work necessary to resolve problems in software design.

This strong belief in free software manifests itself again in the issue of non-free software for documentation. Several BOS core developers discuss the issue of whether to accept documentation from people who did not create it with non-free software. A digest describes the interchange on a mailing list about whether BOS documentation needs to be made with free versus non-free software. Here is an excerpt from someone who objects to the use of Lyx for documentation:

"I really shouldn't have to be harping on this issue for a BOS project, but some ppl like to take convenience over freedom and this should not be tolerated. I mean I would love to read the forms technical reference, but there's no way in hell I am going to install LyX unless I can have a fully free version with the toolkit of my choice (which is supposed to happen eventually)."

This passage has the same tone as the rhetoric from Jim, a believer in the use of free software. Both examples show how the ideology of, and identity with, the open source occupation influences OSSD techniques. Often the ideal of "free" software dominates during negotiations in BOS software development work.

In the BOS project, it appears as though free software guardians review the released code, documentation, and graphics to determine whether people are truly following the beliefs of the FSF. However, the ideology of developing free software according to GPL rights and moral imperatives helps drive conflict resolution. By recording these conversations as IRC logs, all BOS developers and occasional contributors can review and navigate what transpires during virtual project meetings. The IRC logs, email archives, and digests serve as organizational stories that embody the spirit of free software, reinforce its enculturation, and enable its reproduction by and for contributors. We show in the next section how the open disclosure of code and documentation serves as a telling, recording, and retelling of the cultural stories and OSSD narratives revealing work practices to all who are interested in reviewing them (cf. Ryan 2001).

Belief in open disclosure of software development processes

In the "geek" culture, truth is a core priority in developing open source software:

"It should not be too surprising, then, that one of the key values for the community is *truth*. In a world where people are constantly exchanging ideas, evaluating concepts, and suggesting enhancements, it is vitally important that everyone speak the truth as he sees it. If someone fails to speak the truth, the process of creating software will be greatly impaired (Pavlicek, 2000, p. 53)."

In the BOS culture and in the open source culture in general, the importance of speaking the truth in daily work practices is a key value that carries over to OSSD. In the BOS project, this truth element is evident in the belief in the open disclosure of software development processes. This is accomplished by the recording and public archiving of computer-mediated communication via various mediums all recorded for archival purposes: IRC logs, email discussions, and digests. Each digest covers IRC logs and/or email messages for a period of from one to two weeks, includes direct quotes from participants, and includes hyperlinks to the original message sources. A digest reads more like a dramatized account with editorial remarks than like a simple summary of facts. These summaries serve as a resource and organizational memory of activities within the BOS virtual organization (Ackerman and Halverson 2000). Stories are a large part of culture in organizations (Martin, 1992), and here they embody the spirit of the free software movement.

Here is an excerpt from an IRC log discussing a reference to Richard Stallman known in BOS IRC logs as RMS. Previously, someone had talked about RMS as a hero so this discussion is in response to that:

1. <Brandon> Charlie: I should note, on the "RMS hero" thing, that I think it's pretty important to always keep in mind that heroes are regular folks as well.

2. <Charlie> Brandon: I was sorta joking around

3. <Brandon> Charlie: Understood. It's just that a lot of people come up to me at my speeches and conferences and the like and say: "RMS is such a hero; I am amazed at what he's done."

4. <Charlie> I admire him but he's not God

5. <Brandon> Charlie: but I always have to remind them that we can all do what RMS is doing; just by writing Free Software and taking a stand for software freedom. It takes a great and brilliant [mind] to invent the idea of a Free Software Movement and of copyleft, but any one can help those ideas live on.

Most cultures have a leader who is followed (Trice and Beyer, 1992) and RMS is their icon of ideology. By frequent references to Stallman in the IRC logs, those who review them as part of the organizational memory are constantly exposed to the beliefs and values of the FSF.

This belief in disclosing the true methods of software production are manifested in formal and informal work practices. Formal practices include source code check-in/out using CVS, and reporting releases of code via the Web site and mailing list. Informal practices include writing digests to summarize project concerns and development, and providing free access to software development practices in the IRC logs and mailing lists archives.

Freedom of choice in selecting work

Open source developers are attracted to the occupation for its freedom of choice in assignments. Both paid and unpaid BOS participants to some degree can select the work which they prefer. This belief is manifested in the informal methods used to assign and select work in an open source project. The geek culture is known for providing people these choices:

"Another cherished priority in geek culture is the ability of the geek to pursue her passions and ideas. Most people working in the software industry are assigned to projects by their bosses. In geek culture as well, people are often willing to take on tasks that need to be done, even if it is a task they do not relish the thought of pursuing. But geek culture recognizes that there are also tasks that need to be done not because a project requires it, but because the task is burning in the heart and mind of the geek" (Pavlicek, 2000, p. 56).

The BOS data reveals tension between the open source developer's belief in free software and the belief in

freedom of choice when selecting work. For example, in lines 63, Nathan decries the use of non-free software for his graphic exclaiming "I see no reason to avoid non-free software either. If this is really a freedom thing then we should be free to use whatever we want." He believes in freedom of choice yet feels hampered by the strict adherence to using free software to develop an open source system.

ANALYSIS AND DISCUSSION

We now examine the data in terms of three perspectives on occupational culture.

Integration View of BOS Occupational Subculture

From the integration perspective, the content themes of the belief in free software, the belief in open disclosure, and the value of freedom of choice are manifested in the work practices of the virtual organization of open source developers. Contributors, who work closely every day on the IRC lines and via email, yet live in remote locations all over the globe are bonded together by their strong ties to the open source ideology. We have shown in the IRC log that BOS software development practices are driven by an organization-wide consensus towards the goal of free software. All discussions and conflicts revolve around this concept.

As members of an occupational subculture, the developers use each other as reference points - monitoring software development and willingly responding to criticism from a member of the subculture. They form social identities as peers within the BOS subculture building online reputations from their software contributions and extending the participation in IRC lines to after-hours social engagements. On some of the IRC logs, pages of conversational discourse are devoted to playful exchanges, sometimes with people from all over the world communicating during the morning, afternoon, early evening, and early morning hours depending on the time zone. A shared, continually emerging occupational culture binds BOS developers together in their quest to develop a free business enterprise system.

Differentiation View of BOS Subculture

Viewing the BOS occupational culture from the differentiation perspective reveals the formation of subcultures within the BOS group. These subcultures exist in harmony when they are cooperating in software development, and in conflict when they are differing on how software should be developed. The differentiation perspective assumes that conflict is inherent in an organization and attends to the differences. The BOS subculture has two subcultures within it related to free software:

-- a subculture intent on enforcing the sole use of free software for software development of a "free" software product

-- a subculture willing to bend the rules and use non-free software for development and documentation where appropriate and convenient.

In the transcripts, Nathan shows a renegade spirit when discussing free versus non-free software. In the differentiation perspective, people willing to bend the rules relating to the use of non-free software can be viewed as members of a "counter" subculture. While believing in the basic tenants of open source development, these people are comfortable with occasionally using non-free software for development. These two subcultures clash, such as Nathan and Jim did over the graphic, when someone uses non-free instead of free software to produce open source code or documents. However, they are able to resolve the conflict by negotiation and compromise using IRC and mailing lists for communication and mitigation. The overarching identity with the open source occupational subculture creates a bond that facilitates the mitigation of conflict between the two groups.

Fragmentation View of BOS Subculture

The fragmentation perspective focuses on many interpretations that are not evident in the group-wide consensus of the integration view, and that do not focus on the subcultural consensus explicit in the difference perspective (Martin 1992). This perspective assumes ambiguity in the BOS culture rather than excluding it from a cultural analysis. The BOS occupational subculture is constantly in flux with fluid subculture boundaries and no group-wide consensus. From the Fragmentation view, ambiguity is evident in the discussions of free versus non-free software tools and development practices. The protocol for using a non-free versus a free software tool to create code and documentation is unclear. For example, Nathan viewed his Adobe-based diagram as acceptable to the BOS software developers. Probably if Jim had not been "shocked" and dismayed by the existence of this closed source graphic, Nathan would not have changed it. The same holds true for creating BOS documentation using non-free software. Those software developers who believe in the moral imperatives of free software demand that documentation be formed using free tools and those who are grateful for *any* documentation relax their views.

CONCLUSIONS

We have shown how conflicts are communicated and mitigated in a virtual world of software development through the persistent recording of IRC logs and mailing lists. Other researchers have examined online settings where communication between people is constant and immediate. However, this contribution shows the dimensions of work surrounding the recording, Web-based publication and archiving of everyday activity, much like the recording of a conference call, except that the recording is text-based, globally accessible, and referenced by its participants. In the BOS world, people

from various cultures and time zones can work cooperatively at a distance. For them, distance does not matter. They communicate, embody, debate, and resolve conflicts through the use of instant message via IRC, IRC logs, digests, and email archives. In the absence of face-to-face cues and administrative direction, they work cooperatively as a team of developers driven by the mutual desire to develop free software.

We have presented the study of open source software development using the fresh lens of an occupational subculture to show how the belief in developing free software directly influences how software is developed. The use of the three views - Integration, Differentiation, and Fragmentation - portray a richer picture of the occupational subculture than if only one had been used. The preliminary conclusions from this ongoing study include:

-- The recording, publication, archiving, and subsequent referencing of IRC logs assists in conflict resolution. Debates over non-free versus free software are recorded permanently for constant review.

-- Outsiders (lurkers) or occasional contributors can instigate and mitigate organizational conflicts. Code reviews from newcomers and occasional contributors are respected and considered valuable.

-- The three beliefs of the BOS occupational subculture - software freedom principles, freedom of speech, and freedom of choice in work - form a bond within the BOS culture that facilitates the communication and mitigation of conflict in technical choices and work practices.

These three beliefs coalesce to create an inherent belief in computer supported cooperative work in a globally dispersed setting. This belief in cooperative work is a basic underlying assumption (Schein, 1997) in the BOS occupational subculture that exists at a tacit level. The BOS software developers work cooperatively to resolve conflict without formal administrative guidance or informal face-to-face contact.

This research indicates the importance of recorded logs of instant messages for resolving conflicts in virtual work communities. Results have implications for software developers and managers who plan to start an open source project with similar global temporal collocation and virtual communication characteristics. The persistent recording and public archiving of IRC logs, mailing lists, and digests assists open source developers in their daily work and conflict resolution and serves as reminder of the free software philosophy behind open source. Managers of open source projects who are cognizant of the "geek" culture (Pavlicek, 1998) and their fervent belief in the use of free software are better equipped to manage open source software development.

Acknowledgements: This research is funded by grants from the National Science Foundation #IIS-0083075, and

the NSF Industry-University Research Program to the CRITO Consortium at UCI. Mark Ackerman at University of Michigan, Mark Bergman and Xiaobin Li at the UCI Institute for Software Research are collaborators on this research.

REFERENCES

- Ackerman, M. and C. Halverson, Reexamining Organizational Memory, *Communications ACM*, **43**(1), 58-64, Jan 2000.
- Baym, N., From Practice to Culture on Usenet, in S.L. Star, ed., *The Cultures of Computing*, 1996, Blackwell Publishers: Cambridge, MA, 29-52.
- Bergquist, M. and J. Ljungberg, The power of gifts: organizing social relationships in open source communities, *Info. Sys. J.* **11**(4), 305-320, 2001.
- Carmel, E., *Global Software Teams: Collaborating Across Borders and Time Zones*, Prentice-Hall, 1999.
- DiBona, C., S. Ockman, and M. Stone, *Open Sources: Voices from the Open Source Revolution*. 1999, Sebastopol, CA: O'Reilly & Associates Inc.
- Dube, L. and D. Robey, Software Stories: Three Cultural Perspectives on the Organizational Practices of Software Development. *Accounting, Management and Information Technologies*, 1999. **9**(4): p. 223-259.
- Easterbrook, S., ed. *CSCW: Cooperation or Conflict*, 1993, Springer-Verlag: New York.
- Elliott, M., *Organizational Culture and Computer-Supported Cooperative Work in a Common Information Space: Case Processing in the Criminal Courts*. Ph.D. Dissertation. 2000, Information and Computer Science Dept., University of California, Irvine.
- Gregory, K.L., Native-View Paradigms: Multiple Cultures and Culture Conflicts in Organizations, *Administrative Science Quarterly*, **28**, 1983, 359-376.
- Glaser, B. and A.L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. 1967, New York: Aldine.
- Herbsleb, J., A. Mockus, T. Finholt, R. Grinter, Distance, dependencies, and delay in a global collaboration, *Proc. ACM 2000 Conf. Computer Supported Cooperative Work*, December 2000.
- Koch, S. and G. Schneider, Effort, co-operation and coordination in an open source software project: GNOME, *Info. Sys. J.*, **12**(1), 27-42, 2002.
- Martin, J., *Cultures in Organizations: Three Perspectives*, Oxford Univ. Press, 1992.
- Mockus, A., R.T. Fielding, and J. Herbsleb. A Case Study of Open Source Software Development: The Apache Server. in *Proc. Intern. Conf. Software Engineering*, Los Angeles, CA, 263-272, ACM. 2000.
- Muramatsu, J., and M.S. Ackerman, Computing, Social Activity, and Entertainment: A Field Study of a Game MUD, *Comp. Supported Cooperative Work: J. Collaborative Computing*, **7**(1), 87-122, 1998.
- Nardi, B., S. Whittaker, and E. Bradner, Interaction and Outeraction, *Proc. ACM 2000 Conf. Computer Supported Cooperative Work*, 79-88, December 2000.
- Rivett, M., Approaches to Analyzing the Web Text: A Consideration of the Web Site as an Emergent Cultural Form, *Convergence*, **6**(3), Autumn 2000.
- Ryan, M.-L., *Narrative as Virtual Reality: Immersion and Interactivity in Literature and Electronic Media*, Baltimore, MD, John Hopkins Press, 2001.
- Sawyer, S., Effects of Intra-Group Conflict on Packaged Software Development Team Performance. *Info. Sys. J.* 2001. **11**(2), 155-178.
- Scacchi, W. Software Development Practices in Open Software Development Communities, presented at the *1st Workshop on Open Source Software Engineering*, Toronto, Ontario, May 2001.
- Schein, E.H., *Organizational Culture and Leadership*. (Second Edition), 1997, San Francisco: Jossey-Bass.
- Sharma, S., V. Sugumaran, and B. Rajagopalan, A framework for creating hybrid-open source software communities, *Info., Sys. J.*, **12**(1), 7-26, 2002
- Stamelos, I., L. Angelis, A. Oikonomou, and G.L. Bleris, Code Quality Analysis in Open Source Software Development, *Info., Sys. J.*, **12**(1), 43-60, 2002.
- Star, S.L., ed. *The Cultures of Computing*, 1996, Blackwell Publishers: Cambridge, MA.
- Trice, H.M. and J.M. Beyer, *The Cultures of Work Organizations*. 1993, Englewood Cliffs, NJ: Prentice Hall.
- Van Maanen, J.V. and S.R. Barley, Occupational Communities: Culture and Control in Organizations. *Research in Organizational Behavior*, 1984. **6**: 287-365.
- Watson, T., *Sociology, Work & Industry*. 2nd ed. 1987, London: Routledge & Kegan Paul.
- Wenger, E., *Communities of Practice: Learning, Meaning, and Identity*. 1998, Cambridge, Massachusetts: Cambridge University Press.
- Yamaguchi, Y., M. Yokozawa, T. Shinohara, and T. Ishida, Collaboration with Lean Media: How Open Source Software Succeeds, *Proc. Conf. Comp. Supported Cooperative Work (CSCW'00)*, Philadelphia, PA, 329-338, December 2000.