MIS Quarterly

RESEARCH ARTICLE

COLLABORATION THROUGH OPEN SUPERPOSITION: A THEORY OF THE OPEN SOURCE WAY

James Howison
School of Information, University of Texas at Austin, 1616 Guadalupe Avenue,
Austin, TX 78701 U.S.A. {jhowison@ischool.utexas.edu}

Kevin Crowston
School of Information Studies, Syracuse University, 343 Hinds Hall, Syracuse, NY 13244 U.S.A.
and National Science Foundation {crowston@syr.edu}

Appendix A

Coding Scheme for Actions, Inductively Developed

Code	Explanation and Example
Management Codes	
Management	Work done to organize other work. This includes planning, setting deadlines or announcing "phases" like code/string freezes, assigning or rejecting tasks. This includes re-structuring the infrastructure and declaring bugs fixed, or patches applied (closing trackers).
Assigning credit	Thanking people, adjusting the credits file, etc.
Review Codes	
Validation	Validating a coding technique, fix, or approach (before or while it is being done).
Review	Work done to review other work, including checking in code written by others. This includes work that rejects patches, etc.
Production Codes	
Core production	Work that directly contributes to the project's outcomes, either through working application code or through production of user interface elements (logos, etc.); e.g., implementing a feature (not necessarily a check in, since could be checked in on behalf of someone else).
Polishing	Smaller changes that polish core production contributions; e.g., typos, integrations, etc.
Documentation Codes	
Documentation	Work that documents the code, application or activities. Includes pointers across venues (e.g., in a bug tracker saying that a patch has been submitted).
Self-Planning	Work that documents one's own future activities (planning others' work is management work).
Supporting Codes	
Use information provision	Providing or seeking information about using the software; e.g., use cases, often RFEs and bug reports.
Code information provision	Providing or seeking suggestions about the code, including how to complete work (code examples or pseudo-code, if it compiles or is a patch against SVN then code production work). This includes a developer seeking more information from a peripheral member.
Testing	Testing application functionality. This includes requesting more information from users in bug reports.

Appendix B

Summary of Important Terms

Term	Definition
Layer	An outcome of work in the form of a patch that can be applied to an existing, functioning, artifact.
Motivationally independent layer	A layer which, when applied to an existing artifact, provides sufficient motivating payoff without relying on the future completion of other layers.
Solo work	The production of a layer through the programming work of a single person.
Co-work	The production of a layer through the programming work of two or more people, where the motivational payoff for each participant is dependent on the successful completion of all participants' work.
Superposition	The process of laying down motivationally independent layers (thus extending an existing, functioning artifact). Analogous to the geological process of the sedimentary deposition of rock strata on top of another.
Open superposition	A search process led by freely available artifact use resulting in the conceptualization, production and superposition of motivationally and sequentially appropriate layers.
Collaboration through open superposition	A theory of "the open source way." The result of open superposition is that many people have contributed to a functionally interdependent artifact, through the autonomous production of motivationally independent layers. Thus open superposition is a kind of collaboration, even if it did not involve any co-work.

Appendix C

Methodological Appendix

This appendix draws together our methodological approaches for a comprehensive understanding of our method, since our presentation divides our work into three sections. We begin with our general philosophical approach, then discuss our participant observation, our archival replication, and our theory development.

Philosophical Approach

The overall philosophical perspective of this study is one of pragmatism (Diesing 1992; Goldkuhl 2008; Goles and Hirschheim 2000). Rather than beginning with a theoretical problem to be explained, our study began by participating in practice (methodological pragmatism), aiming to first provide a theoretical explanation for why that practice works (referential pragmatism), and finally to demonstrate our theory's usefulness by using it to consider adaptations of this practice (functional pragmatism). These goals informed the overall arc of the research, from deep engagement in the practice leading to theorizing about how and why the practices of FLOSS development work and broadening to asking whether these practices would work in other circumstances. Pragmatism also informed the investigation because constant comparison between participation and the evolving FLOSS literature showed which parts of the lived experience could not be usefully explained by that literature and, therefore, demanded new theorizing. Our theorizing was completed through engagement with practitioners, attempting to use existing theories and our developing theories in productive discussions.

Overall Methodology

The task of this project was always theory development. We were guided by the work of Weick (1989), who describes theory development as *disciplined imagination* directed to *sensemaking* and subject to iterative selection criteria, especially the characteristics of *plausibility and* *interestingness* that develop through this selection process (see below). The selection process itself is realized through *representations*, which themselves require the discipline of writing, familiar from methodologies of ethnography (e.g., van Maanen 1988). In our case, these representations took the form of internal memos, academic publications, and presentations to and conversations with FLOSS developers, through an ongoing active research program, toward an understanding of effective work practices in free and open source software.[1] Below we describe the methodology in more detail for each of the three components of our study, highlighting the ways in which our imagination was disciplined and shaped through iterative selection pressures toward the theory presented in this paper.

Participant Observation

Our case study of BibDesk was an intensive field case study conducted through participant observation, an ethnographic method. The first author was prepared for this through studying case and ethnographic methods in Information Systems (Harvey and Myers 1995; Myers 1999; Yin 1994) and by reading relevant ethnographies (Barley 1986; Knorr-Cetina 1999; Yates and Orlikowski 1992; Zuboff 1988).

Participant observation derives insight from reflection on embedded, longitudinal, lived experience (Lewis 1985; Myers 1999). The first author, therefore, very specifically chose a product that could be integrated into his lived experience as a graduate student (a bibliographic manager). This project spanned four years in the field (exceeding standard recommendations for ethnographic length; Yin 1994, pp. 10-11), cycling up and down for active work episodes, but continually maintaining daily contact with the project through use of the application, as well as subscriptions to mailing lists and bug trackers, in the manner of open source "natives." Contact was limited to online, despite recommendations from some descriptions of participant observation methodology that encourage both interviews and informal social interaction with participants (e.g., Orlikowski 1991). The first author attempted this once, seeking a meeting with the project founder during a trip to San Diego (having noted that he shared an interest in surfing); the negative reply e-mail made it clear that such a social approach was not part of the culture for the project.

Throughout this period, the first author maintained field notes and produced periodic thematic memos for discussion, as recommended by Myers (1999). These notes were frequently discussed with the second author (Walsham and Sahay 1999), as well as a writing group of doctoral colleagues, and through presentations at doctoral consortia. During the analysis phase, the online archives of the project both online and in the e-mail client of the first author assisted in mapping back from field notes to original experiences. For example, not reported in this paper but described in Howison (2009), the observation and experience of primarily individual work was confirmed by a systematic analysis of BibDesk's archival records into episodes, counting aspects such as the number of participants in each different role. This systematic reconstruction functioned as a check on the participant observer's memory (as well as functioning as a pilot study for the archival replication).

The understandings presented in this paper evolved in interaction with two different discourses, disciplining our imagination with selection pressures (Weick 1989). The first was the academic literature grappling with FLOSS. Prior to entering the field, we identified three sensitizing concepts from this literature, discussed in detail in the main body of the paper, and as recommended for qualitative research (Bowen 2006; Glaser 1978; Patton 2002). Then, through our ongoing active research program, we repeatedly explored aspects of the FLOSS phenomena, trying out explanations and incorporating ongoing FLOSS research through a program leading to a FLOSS review paper (Crowston et al. 2012). We sought explanatory cohesiveness, but particularly focused on challenging the literature with lived experience, seeking areas in which they did not correspond. A key guiding principle was to identify and record in memos for discussion the understandings that were surprising in that they were in conflict with the assumptions in the academic literature before that different understanding became commonplace (Myers 1999). An example is the surprising disconnect between the (bursty) temporal rhythm of the lived experience and the flattened time of reading archives for other FLOSS research projects. Another example was the contrast between the focus on interdependency (and its management) in the literature and the experience of aloneness in the conduct and observation of work. A specific example of this occurred when the first author read Yamauchi et al. (2000) and felt it reported an important and little discussed aspect of what he was experiencing (independence), but felt that the explanation (lean media) did not accord with his experience. A second example was developed in conversation with a former Debian project leader who was undertaking FLOSS research as part of his Ph.D. in software engineering, (Michlmayr 2004; Michlmayr and Hill 2003). We discussed the manner in which the volunteer context and the lack of ability to coerce participants was central to discussion about the organization of work among FLOSS participants but strangely de-emphasized in the academic virtual teams literature on FLOSS development.

Simultaneously, we actively engaged with FLOSS practitioners in venues relevant to them, including Apachecon (three times), O'Reilly Open Source (twice), FOOcamp, the Australian Open Source Developers Conference, and LinuxWorld Asia (including a three-day high-level open source leaders speaker's tour to the Taj Mahal that provided occasion for detailed discussion). In these presentations and conversations, we brought academic perspectives (others' and our own evolving ones) and observed the extent to which participants found these *useful* for understanding their own experiences and judged them *interesting* (Davis 1971; Weick 1989). Practitioners, we found, drew blanks when we

[1] http://floss.syr.edu.

discussed purely social or management-driven explanations (such as decision making or leadership), often immediately recognizing these more as features of their "day jobs" that contrasted with their experience in FLOSS work. Instead, they responded with enthusiasm and engagement to explanations that discussed the organization of task work and its motivational context. This repeated pragmatic selection pressure helped draw out the findings presented in our participant observation.

Archival Replication

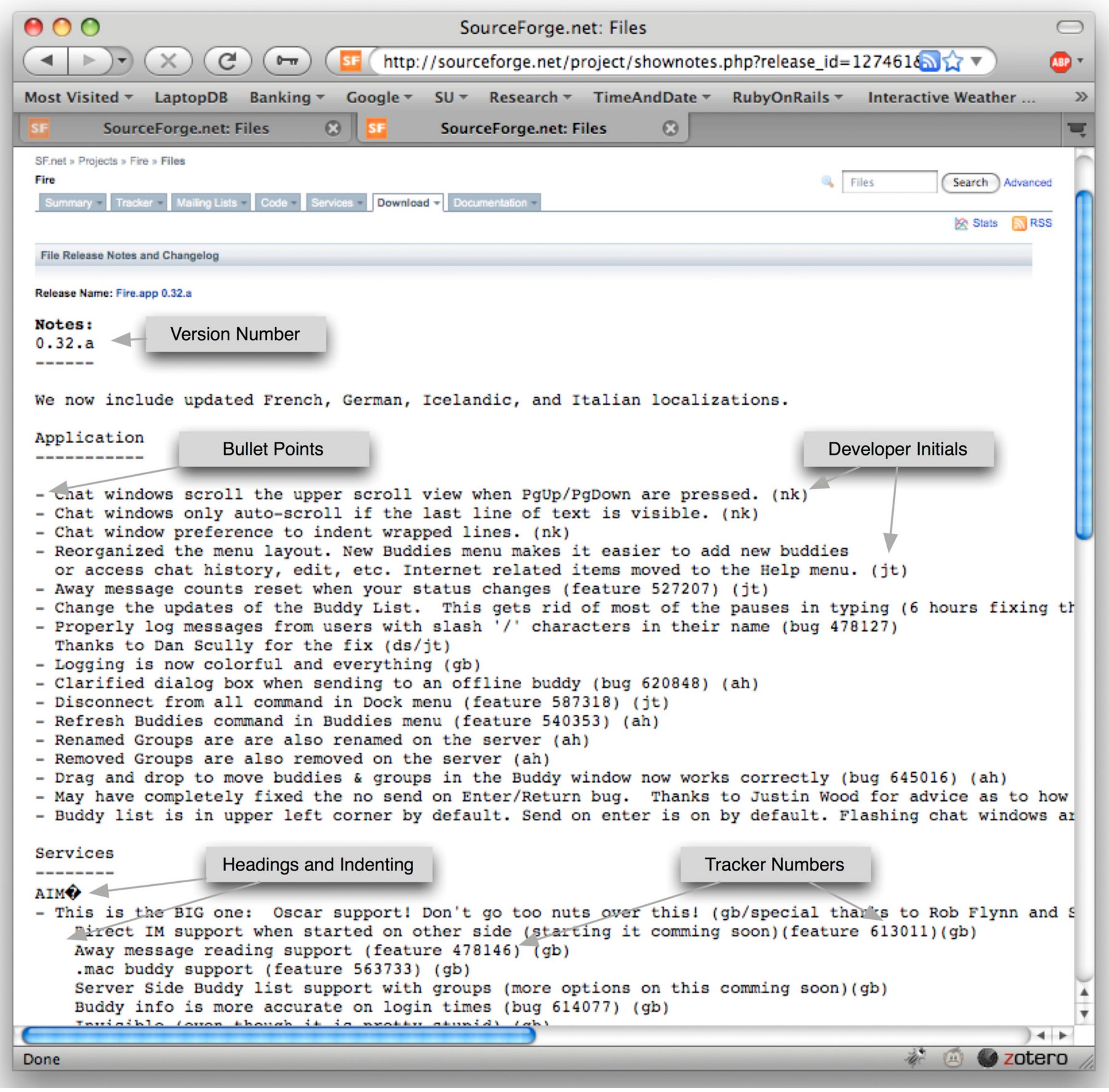
The archival replication is part of the process of theory building, rather than an attempt to verify or test that theory. We argue for its usefulness within that process, highlighting ways in which our sensemaking and imagination were disciplined. The work was undertaken to build confidence that the processes observed in our case study did not derive only from the specifics of our single case study and thus proved sufficiently disciplined for useful theorizing.

The reconstruction of the work episodes from archival records relies on a method which Geiger and Ribes (2011) would later call "trace ethnography," which involves a process of inversion (Bowker and Star 2000) that connects back from digital trace records to lived experience. This process draws on an ethnographic understanding of the "sociotechnical infrastructure of documentary practices" (Geiger and Ribes 2011) that is then drawn on in interpreting the archival records. Thus, within the overall process of theory development, the ethnographic understandings from participant observation were important in the archival reconstruction work conducted by the first author, disciplined as described below. The categories employed in this analysis are somewhat idealized, but are grounded in how developers think about the world, shown by the manner in which they contribute task outcomes to the release notes.

The overall purpose of the archival analysis was narrative reconstruction, focusing particularly on sequence and focal actors (Pentland 1999). There are two analytic moves in the reconstruction: the identification of tasks and the classification of actions within those tasks. This process was disciplined in three ways. First, tasks were identified through FLOSS team members' own records of the outcomes of their work, through individual bullets added to the release notes over time, and CVS log messages; these source are emic and form the anchors for the task. Second, the codebase itself, as a well-structured artifact, provided discipline: the analyst was able to trace a source code change in context, comparing between versions and tracing paths of execution (see Figure C2). Third, the analysis sought narrative cohesion, relying on the necessary logic of sequence: that some actions must precede others (e.g., writing code must proceed its review) (see Pentland 1999). These three principles discipline the analysis, resulting in experimental splits and merges of tasks, seeking narrative coherence.

The identification of the actions as of different types was conducted by the first author, disciplined by both narrative cohesion and, although a formal coding reliability test was not performed, an outside colleague's review of a sample of the tasks with their coded actions, confirmed narrative cohesion. The key finding (of few tasks in which more than one programmer writes code) relates to the identity of those checking code into CVS as recorded by the system, providing the fourth source of discipline. The analyst merely linked CVS user names and e-mail addresses across identities, drawing on database fields and hints in the textual data (such as initials, see Figure C1; see also Geiger and Ribes 2011).

The figures convey the interpretative apparatus. Figure C1 shows the release notes, providing the key anchors for the recognition of tasks. Figure C2 shows the Sourceforge CVS diff browser that enables understanding the code changes in context, browsing linked files and following lines of code execution. Figure C3 shows the interpretative coding apparatus (using RDF expressed in turtle triples); note the memos (relevance_memo, task_memo and ca_memo, a code application memo). The structured database constructed in this manner was then queried in two ways, the first to output each task in its sequence for further interpretation (as in Table 2 in the main body of the paper), the second to output counts of event attributes (such as number of programmers or length of task) used to report quantitative results and create the graphs shown in this paper.

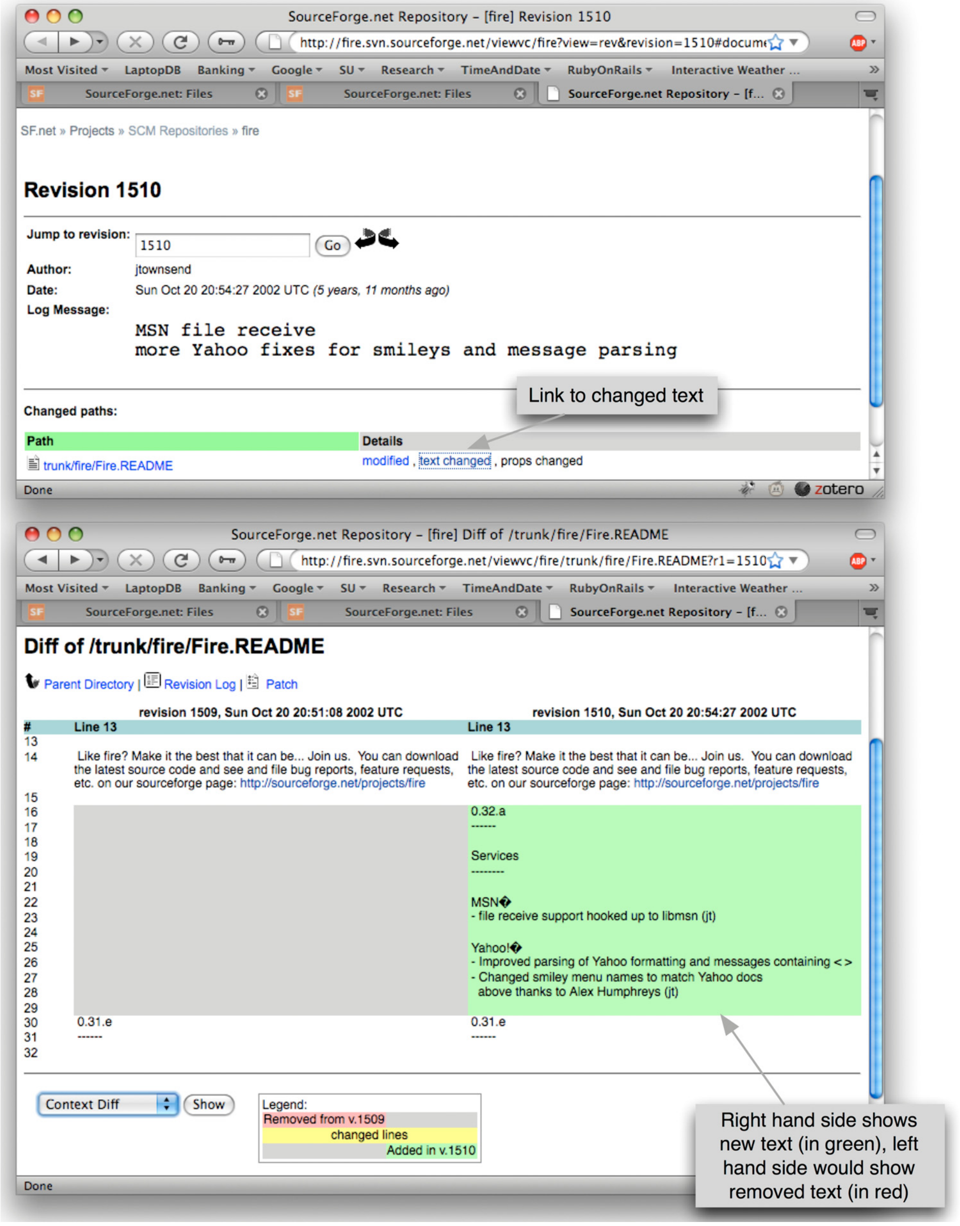


File Release Notes and Changelog

Release Name: Fire.app 0.32.a

```
Notes:
0.32.a
------

We now include updated French, German, Icelandic, and Italian localizations.

Application
-----------

- Chat windows scroll the upper scroll view when PgUp/PgDown are pressed. (nk)
- Chat windows only auto-scroll if the last line of text is visible. (nk)
- Chat window preference to indent wrapped lines. (nk)
- Reorganized the menu layout. New Buddies menu makes it easier to add new buddies
  or access chat history, edit, etc. Internet related items moved to the Help menu. (jt)
- Away message counts reset when your status changes (feature 527207) (jt)
- Change the updates of the Buddy List.  This gets rid of most of the pauses in typing (6 hours fixing th
- Properly log messages from users with slash '/' characters in their name (bug 478127)
  Thanks to Dan Scully for the fix (ds/jt)
- Logging is now colorful and everything (gb)
- Clarified dialog box when sending to an offline buddy (bug 620848) (ah)
- Disconnect from all command in Dock menu (feature 587318) (jt)
- Refresh Buddies command in Buddies menu (feature 540353) (ah)
- Renamed Groups are are also renamed on the server (ah)
- Removed Groups are also removed on the server (ah)
- Drag and drop to move buddies & groups in the Buddy window now works correctly (bug 645016) (ah)
- May have completely fixed the no send on Enter/Return bug.  Thanks to Justin Wood for advice as to how
- Buddy list is in upper left corner by default. Send on enter is on by default. Flashing chat windows ar

Services
--------
AIM�
- This is the BIG one:  Oscar support! Don't go too nuts over this! (gb/special thanks to Rob Flynn and S
   Direct IM support when started on other side (starting it comming soon)(feature 613011)(gb)
   Away message reading support (feature 478146) (gb)
   .mac buddy support (feature 563733) (gb)
   Server Side Buddy list support with groups (more options on this comming soon)(gb)
   Buddy info is more accurate on login times (bug 614077) (gb)
   Invisible (even though it is pretty stupid) (gb)
```

Figure C1. Release Notes Providing Task Outcomes as Anchors for Task Reconstruction



SourceForge.net Repository – [fire] Revision 1510
http://fire.svn.sourceforge.net/viewvc/fire?view=rev&revision=1510#docume
Most Visited
LaptopDB
Banking
Google
SU
Research
TimeAndDate
RubyOnRails
Interactive Weather ...
SourceForge.net: Files
SourceForge.net: Files
SourceForge.net Repository – [f...

SF.net » Projects » SCM Repositories » fire

Revision 1510

Jump to revision:	1510 Go
Author:	jtownsend
Date:	Sun Oct 20 20:54:27 2002 UTC *(5 years, 11 months ago)*
Log Message:	

```
MSN file receive
more Yahoo fixes for smileys and message parsing
```

Link to changed text

Changed paths:

Path	Details
trunk/fire/Fire.README	modified , text changed , props changed

Done
zotero
SourceForge.net Repository – [fire] Diff of /trunk/fire/Fire.README
http://fire.svn.sourceforge.net/viewvc/fire/trunk/fire/Fire.README?r1=1510
Most Visited
LaptopDB
Banking
Google
SU
Research
TimeAndDate
RubyOnRails
Interactive Weather ...
SourceForge.net: Files
SourceForge.net: Files
SourceForge.net Repository – [f...

Diff of /trunk/fire/Fire.README

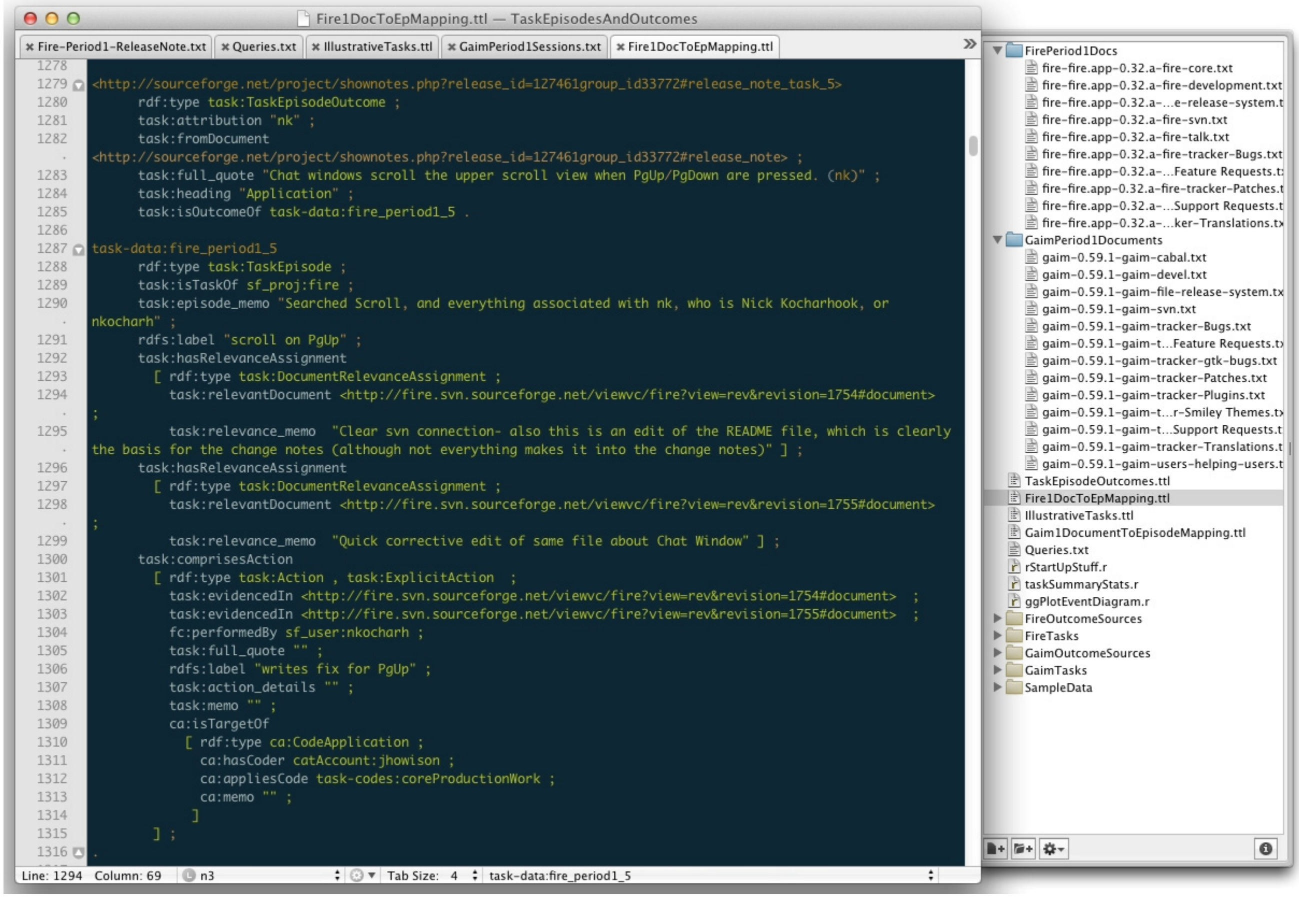
Parent Directory | Revision Log | Patch

	revision 1509, Sun Oct 20 20:51:08 2002 UTC	revision 1510, Sun Oct 20 20:54:27 2002 UTC
#	Line 13	Line 13
13		
14	Like fire? Make it the best that it can be... Join us. You can download the latest source code and see and file bug reports, feature requests, etc. on our sourceforge page: http://sourceforge.net/projects/fire	Like fire? Make it the best that it can be... Join us. You can download the latest source code and see and file bug reports, feature requests, etc. on our sourceforge page: http://sourceforge.net/projects/fire
15		
16		0.32.a
17		------
18		
19		Services
20		--------
21		
22		MSN�
23		- file receive support hooked up to libmsn (jt)
24		
25		Yahoo!�
26		- Improved parsing of Yahoo formatting and messages containing < >
27		- Changed smiley menu names to match Yahoo docs
28		above thanks to Alex Humphreys (jt)
29		
30	0.31.e	0.31.e
31	------	------
32		

Legend:
🟥 Removed from v.1509
🟨 changed lines
🟩 Added in v.1510

Right hand side shows new text (in green), left hand side would show removed text (in red)

Figure C2. The Sourceforge CVS Browser, Facilitating Knowledge of Code in Context


Fire1DocToEpMapping.ttl — TaskEpisodesAndOutcomes
Fire-Period1-ReleaseNote.txt
Queries.txt
IllustrativeTasks.ttl
GaimPeriod1Sessions.txt
Fire1DocToEpMapping.ttl
Line: 1294 Column: 69
n3
Tab Size: 4
task-data:fire_period1_5
FirePeriod1Docs
fire-fire.app-0.32.a-fire-core.txt
fire-fire.app-0.32.a-fire-development.txt
fire-fire.app-0.32.a-...e-release-system.t
fire-fire.app-0.32.a-fire-svn.txt
fire-fire.app-0.32.a-fire-talk.txt
fire-fire.app-0.32.a-fire-tracker-Bugs.txt
fire-fire.app-0.32.a-...Feature Requests.t:
fire-fire.app-0.32.a-fire-tracker-Patches.t
fire-fire.app-0.32.a-...Support Requests.t
fire-fire.app-0.32.a-...ker-Translations.tx
GaimPeriod1Documents
gaim-0.59.1-gaim-cabal.txt
gaim-0.59.1-gaim-devel.txt
gaim-0.59.1-gaim-file-release-system.tx
gaim-0.59.1-gaim-svn.txt
gaim-0.59.1-gaim-tracker-Bugs.txt
gaim-0.59.1-gaim-t...Feature Requests.tx
gaim-0.59.1-gaim-tracker-gtk-bugs.txt
gaim-0.59.1-gaim-tracker-Patches.txt
gaim-0.59.1-gaim-tracker-Plugins.txt
gaim-0.59.1-gaim-t...r-Smiley Themes.tx
gaim-0.59.1-gaim-t...Support Requests.t
gaim-0.59.1-gaim-tracker-Translations.t
gaim-0.59.1-gaim-users-helping-users.t
TaskEpisodeOutcomes.ttl
Fire1DocToEpMapping.ttl
IllustrativeTasks.ttl
Gaim1DocumentToEpisodeMapping.ttl
Queries.txt
rStartUpStuff.r
taskSummaryStats.r
ggPlotEventDiagram.r
FireOutcomeSources
FireTasks
GaimOutcomeSources
GaimTasks
SampleData

```

<http://sourceforge.net/project/shownotes.php?release_id=127461group_id33772#release_note_task_5>
      rdf:type task:TaskEpisodeOutcome ;
      task:attribution "nk" ;
      task:fromDocument
<http://sourceforge.net/project/shownotes.php?release_id=127461group_id33772#release_note> ;
      task:full_quote "Chat windows scroll the upper scroll view when PgUp/PgDown are pressed. (nk)" ;
      task:heading "Application" ;
      task:isOutcomeOf task-data:fire_period1_5 .

task-data:fire_period1_5
      rdf:type task:TaskEpisode ;
      task:isTaskOf sf_proj:fire ;
      task:episode_memo "Searched Scroll, and everything associated with nk, who is Nick Kocharhook, or
nkocharh" ;
      rdfs:label "scroll on PgUp" ;
      task:hasRelevanceAssignment
        [ rdf:type task:DocumentRelevanceAssignment ;
          task:relevantDocument <http://fire.svn.sourceforge.net/viewvc/fire?view=rev&revision=1754#document>
;
          task:relevance_memo  "Clear svn connection- also this is an edit of the README file, which is clearly
the basis for the change notes (although not everything makes it into the change notes)" ] ;
      task:hasRelevanceAssignment
        [ rdf:type task:DocumentRelevanceAssignment ;
          task:relevantDocument <http://fire.svn.sourceforge.net/viewvc/fire?view=rev&revision=1755#document>
;
          task:relevance_memo  "Quick corrective edit of same file about Chat Window" ] ;
      task:comprisesAction
        [ rdf:type task:Action , task:ExplicitAction  ;
          task:evidencedIn <http://fire.svn.sourceforge.net/viewvc/fire?view=rev&revision=1754#document>  ;
          task:evidencedIn <http://fire.svn.sourceforge.net/viewvc/fire?view=rev&revision=1755#document>  ;
          fc:performedBy sf_user:nkocharh ;
          task:full_quote "" ;
          rdfs:label "writes fix for PgUp" ;
          task:action_details "" ;
          task:memo "" ;
          ca:isTargetOf
            [ rdf:type ca:CodeApplication ;
              ca:hasCoder catAccount:jhowison ;
              ca:appliesCode task-codes:coreProductionWork ;
              ca:memo "" ;
            ]
        ] ;
.
```

Figure C3. The Interpretative Apparatus, Producing Structured RDF Linking Archives, Memos, Codes, Actors, and Event Times

Theory Building

Weick's method for theory building relies on selection pressures toward "plausibility and interestingness," echoing the criterion of *plausibility* for ethnographic work discussed by Myers (1999) and Prasad (1997). Earlier, we described the iterative ongoing practice of working with both the academic literature and practitioners that provided the selection pressures. However, the selection characteristic of *interestingness* is potentially diffuse. Here we were guided primarily by Davis (1971), who argued that "interesting theories deny...weakly-held assumptions of their audience" (as quoted by Van de Ven 2007, p. 111). In our case, we deny two weakly held assumptions of the literature (rather, we hope we deny them and we hope they are only weakly held).

The first is an assumption of the virtual teams literature that FLOSS projects had found a way to coordinate and organize to accomplish complex work, despite the challenges of high interdependency of software work, working with volunteers, and working at distance. In contrast, we found that FLOSS projects simply chose not to undertake complex work but to defer it. Moreover, we found that deferring such complex work is key to the open source way of working.

The second assumption was that the core of the open source way of working was not closely bound to software work, and would thus relatively easily yield adaptable explanations (such as creation of trust, leadership, or decision-making techniques). In contrast, we found that software work (or rather, particular characteristics of software work that might conceivably be found elsewhere) were central to explaining the conundrum of how not doing complex work can ever lead to a collective complex artifact.

These aspects of interestingness, however, did not develop separately nor did they develop alone. Rather, we tried many explanations, both in conversation and in writing to both practitioners and academics, en route to these explanations. In particular, we returned to field notes and memos to choose and write the vignettes that illustrate our participant observation. In a second phase, through conversation with colleagues, reviewers, and multiple iterations, we confronted the need to face the question logically posed by the implications of the participant observation and archival reconstruction (How does mostly individual work result in a complex artifact?), clarifying the theory of collaboration through open superposition.

Finally, our pragmatic philosophical approach called for a demonstration of the usefulness of this theory. This we undertook through the analysis of the contingencies for the success of the theorized way of working and thereby casting light on the limits of the adaptability of the open source way. In these ways, our imagination was disciplined toward the theory presented in this paper.

References

Barley, S. R. 1986. "Technology as an Occasion for Structuring: Observations on CT Scanners and the Social Order of Radiology Departments," *Administrative Science Quarterly* (31:1), pp. 78-108.

Bowen, G. A. 2006. "Grounded Theory and Sensitizing Concepts," *International Journal of Qualitative Methods* (5:3), pp. 12-23.

Bowker, G. C., and Star, S. L. 2000. *Sorting Things Out: Classification and Its Consequences*, Cambridge, MA: MIT Press.

Crowston, K., Wei, K., Howison, J., and Wiggins, A. 2012. "Free (Libre) Open Source Software Development: What We Know and What We Do Not KNow," *ACM Computing Surveys* (44:2), Article 7.

Davis, M. S. 1971. "That's Interesting!," *Philosophy of Social Science* (1:4), pp. 309-344.

Diesing, P. 1992. *How Does Social Science Work? Reflections on Practice*, Pittsburgh PA: University of Pittsburgh Press.

Geiger, R. S., and Ribes, D. 2011. "Trace Ethnography: Following Coordination through Documentary Practices," in *Proceedings of the 44th Hawaii International Conference on System Sciences*, Los Alamitos, CA: IEEE Computer Society Press.

Glaser, B. G. 1978. *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory*, Mill Valley, CA: Sociology Press.

Goldkuhl, G. 2008. "What Kind of Pragmatism in Information Systems Research?," in *Proceedings of AIS Special Interest Group on Pragmatist IS Research*, Paris, France, December 14, pp. 3-8.

Goles, T., and Hirschheim, R. 2000. "The Paradigm Is Dead, the Paradigm Is Dead…Long Live the Paradigm: The Legacy of Burrell and Morgan," *Omega* (28:3), pp. 249-268.

Harvey, L. J., and Myers, M. D. 1995. "Scholarship and Practice: The Contribution of Ethnographic Research Methods to Bridging the Gap," *Information Technology & People* (8:3), pp. 13-27.

Howison, J. 2009. "Alone Together: A Socio-Technical Theory of Notivation, Coordination and Collaboration Technologies in Organizing for Free and Open Source Software Development," unpublished Ph.D. dissertation, Syracuse University.

Knorr-Cetina, K. 1999. *Epistemic Communities*, Cambridge, MA: Harvard Education Press.

Lewis, I. M. 1985. *Social Anthropology in Perspective: The Relevance of Social Anthropology*, Cambridge, UK: Cambridge University Press.

Van Maanen, J. 1988. *Tales of the Field: On Writing Ethnography*, Chicago: University of Chicago Press.

Michlmayr, M. 2004. "Managing Volunteer Activity in Free Software Projects," in *Proceedings of the 2004 USENIX Annual Technical Conference, FREENIX Track*, Boston, June 27-July 2, pp. 39-50.

Michlmayr, M., and Hill, B. M. 2003. "Quality and the Reliance on Individuals in Free Software Projects," in *Proceedings of the 3rd Workshop on Open Source Software Engineering*, Cork, Ireland, May 3-10, pp. 785-786.

Myers, M. 1999. "Investigating Information Systems with Ethnographic Research," *Communications of the Association for Information Systems* (2), Article 23.

Orlikowski, W. J. 1991. "Integrated Information Environment or Matrix of Control? The Contradictory Implications of Information Technology," *Accounting, Management and Information Technologies* (1:1), pp. 9-42.

Patton, M. Q. 2002. *Qualitative Research and Evaluation Methods*, London: Sage Publications.

Pentland, B. T. 1999. "Building Process Theory with Narrative: From Description to Explanation," *Academy of Management Review* (24:4), pp. 711-724.

Prasad, P. 1997. "Systems of Meaning: Ethnography as a Methodology for the Study of Information Technologies," in *Information Systems and Qualitative Research*, A. S. Lee, J. Liebenau, and J. I. DeGross (eds.), London: Chapman & Hall, pp. 101-118.

Van de Ven, A. 2007. *Engaged Scholarship: A Guide for Organizational and Social Research*, Oxford, UK: Oxford University Press.

Walsham, G., and Sahay, S. 1999. "GIS for District-Level Administration in India: Problems and Opportunities," *MIS Quarterly* (23:1), pp. 39-65.

Weick, K. E. 1989. "Theory Construction as Disciplined Imagination," *Academy of Management Review* (14:4), pp. 516-531.

Yamauchi, Y., Shinohara, T., and Ishida, T. 2000. "Collaboration with Lean Media: How Open-Source Software Succeeds," in *Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work*, Philadelphia, Pennsylvania, PA, December 2-6, pp. 329-338.

Yates, J., and Orlikowski, W. J. 1992. "Genres of Organizational Communication: A Structurational Approach to Studying Communication and Media," *The Academy of Management Review* (17:2), pp. 299-326.

Yin, R. 1994. *Case Study Research: Design and Methods* (2md ed.) Newbury Park, CA: Sage Publications.

Zuboff, S. 1988. *In the Age of the Smart Machine: The Future of Work and Power*, New York: Basic Books.