What characterize documents that bridge boundaries compared to documents that do not? An exploratory study of documentation in FLOSS teams

Carsten Østerlund
Syracuse University
costerlu@syr.edu

Kevin Crowston
Syracuse University
crowston@syr.edu

Abstract

Organizations bring together people with various access to and understanding of the work at hand. Despite their different stocks of background knowledge, most of them engage in documentation, whether as writers or readers. This paper explores how documents serve such diverse users by building a framework articulating the characteristics of documents supporting collaborators with asymmetric access to knowledge versus people with symmetric knowledge. Drawing on document-centric approaches we hypothesize that documents supporting asymmetric groups are likely to be more prescriptive and explicate their own use compared to documents supporting symmetric groups. Through exploratory analysis of two kinds of documents, used across three FLOSS projects, we find that documents supporting collaborators with asymmetric knowledge do appear to explicate their own use in more detail. They do so by prescribing their own 1) purpose, 2) context of use, 3) content and form in greater detail than documents used by core community members with symmetric access to project knowledge.

1. Introduction

Most organizational endeavors require extensive documentation to facilitate communication, coordination, and for accounting purposes. In distributed environments the use of documents become even more prevalent, often constituting the only means for interaction among collaborators. Yet, documenting work is complicated by the fact that organizational members often bring divergent understandings and knowledge to the production and use of documents. Collaborators may in some cases draw on highly congruent stocks of knowledge, for example, if they have worked together for a long time and hold comparable social positions. If so, a few words uttered can be enough to convey the status of a shared project. But people may bring different stocks of background knowledge to their social interactions as access to knowledge may be unevenly distributed among members of a community and can differ substantially from community to community. Collaborators may bring different frames of reference that do not converge and share only a general sense of a project, an understanding that is relatively more vague and anonymous than that of consociates. In such case, a few words uttered by a core member to a peripheral participant would be insufficient to convey the scope of a project let alone its status.

If we cannot take on a Saussurian and Chomskyan idealization that all speaker-listeners know their language identically, one has to approach a document as a dynamic entity that stands in an active relationship to its context. In other words, documents do not simply deliver certain sort of readings that are forces upon the reader. Instead, the reader engages with documents in a reciprocal relationship where the one informs the other.

Documents interact with their audience by acting as prescriptive devices [1-4]. They denote their use by telling the reader how they, the documents are to be applied and the proper context of their use. The producers of a document do so by bringing specific parts of their world into focus through the content of the documents and its assumed context of use. In this way, documents account for something. They are symbols that model the empirical reality by depicting physical, organize, social or psychological systems by paralleling, imitating or simulating them.

The notion of account though has a double sense: an “of” and a “for” sense [5]. On the one hand, documents provide an account "of" reality by manipulating text and other symbolic structures so as to parallel them with reality. Software engineers may carefully document the code they have constructed in an open source project by creating a report of the work done. By expressing the code in a synoptic form that renders it apprehensible, the software engineers create a model “of reality.”

On the other hand, an account may serve as the basis on which people manipulate the world. The engineers’ report is not simply an account of work completed: the report may guide the ongoing work by prescribing what is left to be done or allow collaborators to coordinate their ongoing work. The report may be used when trying to fix a bug in the program. It is an

account “for reality” as it provides a blueprint of the software program taking shape. Documents thus offer a double accountability: when documenting the coding of a software program, engineers mold the account to the reality of the code on their computers and at the same time, mold their ongoing coding to the account.

However, people are able to recognize a document's prescriptions (i.e., accounts of and for work) only if they are familiar with the document genre and the social field in which these documents exist. Their background knowledge enables them to understand how to use the document. The possibility of understanding the document is based on some common point of reference between the writer and reader. Users of documents need to know how those documents are bound up with certain kinds of organizational practices [6]. They need to have the background knowledge, or a set of expectations and assumptions that enable them to approach the documents intelligently. A novice programmer with no history in the particular project may be able to read the report and understand its content at a sentence or paragraph level and get some sense of the work completed, but remain unable to use the software engineers’ report as an account for work. With little knowledge of the organizational practices that went into the creating of the code and the report the novice is left frustrated and confused with little sense of what to do next.

The dynamic relationship between documents as prescriptive devices and users as carriers of background knowledge create an interesting dynamic: how much background knowledge do the users bring and how much does the document explicate? For instance, if the writer and reader share little background knowledge then the document facilitating their communication will likely explicate its use in great detail. Conversely, if the collaborators have a long history working together and hold a common point of reference their documents do not need to go into as much detail. Their intentions can be read out of the slightest hints and references to past experiences. In short, the dynamic relationship between documents and users depends on the level of symmetric access to knowledge about the writers and readers’ context. When we turn our attention to the documents in this dynamic relationship, we are left with the question:

> What characterize documents that link people with asymmetric access to background knowledge compared to documents mediating communication among people with symmetric access to knowledge?

The purpose of this paper is to develop a theoretical perspective on documents that helps answer this question and to illustrate our perspective with examples from a particular setting.

2. Theory elaboration and propositions

A review of the literature offers few specific answers to our question. It is difficult to find scholars detailing specific linguistic or textual features that distinguish documents’ prescriptive elements. Institutional ethnography, for instance, as articulated by Dorothy Smith, defines documents’ self-explicating capabilities but describes the dynamic in broad strokes [1, 6]. To mitigate this lack of detail we turn to two bodies of work for help: genre theory and Star and Bowker’s work on boundary objects and classification. The first focuses on the common stock of knowledge people bring to their document production and use. The second pays attention to documents and other artifacts that bridge people with little shared point of reference. We will address these in turn.

2.1 Genre theory

Rhetoricians since Aristotle have attempted to classify documents into categories or “genres” with similar form, topic or purpose. More recently a document genre has been defined as typified communicative action invoked in response to a recurrent situation [7-9]. People engage genres to accomplish social actions in particular situations, characterized by a particular purpose, content, form, and set of participants in specific times and places. Viewed from the perspective of the writer and reader, identification of a document’s genre makes the document more easily recognizable and understandable, thus reducing the amount of detail required to convey meaning. For genres to be of aid in communication though, they must be shared across the members of particular discourse community [10]. Thus, genres depend on symmetric access to knowledge among a group of people. A genre may be unfamiliar or hard to understand for someone outside a community but readily accessible to an insider. Community members familiar with a genre are likely to know the expectations. Conversely, people with little access to the background knowledge of the community are not likely to know the genre and in turn bring few if any expectations about what purpose, content, and form a document in that genre is likely to convey and what set of participants have produced and use it at what times and places.

Following this line of argument we can assume that people with symmetric access to knowledge do not need to explicate those genre expectations, but can lean heavily on them when writing and reading documents. A hint by one core developer of an open source project to another core developer may be enough to invoke a genre. In contrast, if we need to facilitate communication among people with asymmetric access to

knowledge the documents would have to articulate those communicative expectations for them. That is, the document would be more likely to explicitly state its purpose, form, content, appropriate participants, and time and place of the communication. We can break this process into three general areas: The document explicates: 1) its *purpose*; 2) the elements that go into the document itself in terms of *form* and *content;* and 3) the context of its use in terms of *participants*, *times* and *places*? These considerations lead us to three general hypotheses:

Hypothesis 1: A document shared among people with asymmetric knowledge is more likely to explicitly state its purpose.

Hypothesis 2: A document shared among people with asymmetric knowledge is more likely to explicate the context of use in regard to appropriate participants, times and places of its production and use.

Hypothesis 3: A document shared among people with asymmetric knowledge is more likely to explicate the form and content of its communication.

2.2 Boundary objects theory

In an attempt to further specify these hypotheses we turn to Star and Bowker's work concerning communicative practices in situations characterized by heterogeneity [11, 12]. Actors from different communities, with little shared point of reference and common stock of knowledge have to manage the tension between their divergent viewpoints. Star introduces the concept of boundary object to explain how such heterogeneous communities maintain a productive communication despite their desperate perspectives and asymmetric access to knowledge. Following this line of though, we can assume that documents typically shared among groups with asymmetric access to knowledge may take the form of boundary objects. If so, the boundary object literature could hint at what characterize such documents in terms of their prescriptive and self-explicating features.

Star highlights four types of boundary objects: repository, coincidence boundary, ideal type, and standardized form. Repositories constitute collections of documents or other artifacts and thus are not relevant for our present discussion. The remaining three types offer some helpful hints. Coincidence boundaries and ideal type objects point to how documents can explicate the context of their use whereas standardized forms suggest how documents may prescribe their content and form.

Context of use: Star defines *coincidence boundaries* as common objects that have the same boundaries but different internal content. They arise when work is distributed over a large-scale geographic area. Star points to the state of California as a coincidence boundary for the collaboration among citizen scientists and professional biologists at UC Berkeley. The map of California explicates the common context for the different groups' collaboration and communication. The result is that work in different sites and with different perspectives can be conducted autonomously while cooperating parties share a common spatial referent. Extending Star's thinking one could imagine documents that specify e.g., temporal or participatory boundaries.

Ideal types are documents such as diagrams, atlas, or other descriptions that in fact do not accurately describe the details of any one locality, thing or activity. They may be fairly vague and abstract. However, it is this very quality which makes it useful to people with different points of reference and stock of knowledge precisely. It offers a good enough road map to demarcate general elements, processes or organization of the shared context. This argument suggest that people with symmetric access to knowledge, such as two core developers do not need to use ideal type documents, such as diagrams, to facilitate their communication and collaboration. They do not need a road map: they know the way. However, people who share little common stock of knowledge and exist at the periphery of the community may need diagrams and road maps to navigate and be able to read and use a document.

Coincidence boundaries and ideal types allow us to further articulate our second hypothesis:

Hypothesis 2: A document shared among people with asymmetric knowledge is more likely to explicate the context of use:

A. By specifying the appropriate participants, times and places of its production and use
B. Through ideal types, such as diagrams, atlas, road maps, which demarcate the specific elements or organization of the shared work.
C. By demarcating the boundaries of the shared work. These can be geographical space or other specific boundaries about the scope of the work required by the project and the specific document.

Content and form: Standardized forms include labels and other documents offering a uniform way to index communicative content and form. While Star highlights how standardized forms delete local uncertainties from the shared information, one could also turn this point on its head and show that the standardized forms in fact articulate a basic structure for the document's content and form. This might not be a very detailed prescription but it nevertheless specifies the in-

formation needed for the particular communicative relationship supported by the document. People with intimate knowledge of the work at hand have less need for standardized forms. They know what they have to get done and what information will be relevant to the work at hand.

In more recent work, Bowker and Star extend their interest in the communication among heterogeneous groups to classification schemes [12]. Documents that are used to account for work will often draw on some classification scheme of that work. One could argue that a standardized form builds on some classification system. Two issues become particular important when understanding classification systems designed for heterogeneous groups: *comparability* and *visibility*. These look different close to the work situation and at a distance. People with symmetric access to knowledge do not have the same requirements for comparability and visibility as somebody with asymmetric access to knowledge.

A major purpose for creating accounts in a document is to provide a good comparability across descriptions to enhance communication. For people to use work accounts some regularity in semantics and objects from one document to the other must exist. The more intimate the producers and users of documents, the less necessary such schemes need to be. If people know the situation and practices in detail very little need to be documented for them to make use of the text. In contrast, if people have little background knowledge the document would have to prescribe the regularity in semantics and objects required to facilitate comparison across documents.

The same dynamic plays out when it comes to visibility. When accounting for work activities one must differentiate areas of work that are invisible and visible. It would be unproductive, if even possible, to record everything. Some work just gets done with no need to document, with no voice in the accounting scheme. Invisibility can be regarded as erasure or just not important. But invisibility can also stem from intimacy as in a team that has worked together for so long that they no longer need to account for certain activities. They are simply assumed, while for people at some distance, the document and its classification scheme would have to require more detailed description of activities.

Based on the notions of standardized forms and the visibility and comparability of document content we can refine our third hypothesis:

Hypothesis 3: A document shared among people with asymmetric knowledge is more likely to explicate the form and content of its communication by:

A. Bringing regularity in semantics and objects covered by one document to the next, e.g., through standardized forms that offer structured way to index communicative content.
B. Requiring the users to make more details of their work visible in their descriptions.

3. Method and setting

To illustrate the hypotheses developed above, we sought a setting in which we could observe documents being used across groups with different kinds and levels of shared background knowledge. We chose to study documents used in Free/Libre/Open Source Software (FLOSS) development. Key to our interest is the fact that most FLOSS projects are developed by virtual teams comprising professionals and users [13, 14]. These teams are close to purely virtual in that developers contribute from around the world, meet face-to-face infrequently if at all and coordinate their activity primarily by means of a variety of computer-mediated communication (CMC) tools [15, 16]. The scarcity of face-to-face interactions seems to be balanced by substantial use of communications channels available online [17]. As development proceeds, evidence of the processes and interactions between tasks and participants is left in repositories such as email lists, issue trackers, source code management systems and so on. These channels are characterized by documents of different genres that make up the FLOSS genre repertoire. For example, bug trackers have genres that are constrained by the features of the system. Email is more flexible, but still shows distinct patterns of use. Most project maintain a variety of on-line documents in specific and genres, such as manuals, FAQs, to do lists and help. Finally, the source code itself is a powerful communication medium in these teams.

A particular interest is how the use of these varied documents reveals patterns of relationships among different kinds of members of a FLOSS team. Several authors have described successful FLOSS teams as having a hierarchical [18] or onion-like structure [19-22]. At the centre are the *core developers*, who contribute most of the code and oversee the design and evolution of the project. Core developers are usually distinguished by having write privileges or other formal authority over the source code [23, 24]. Core developers contribute most of the code and oversee the design and evolution of the project. Most core developers know and acknowledge each other's contributions. The core is usually small, as there is a high level of interaction among core members, which would be difficult to maintain if the core were large. Core developers communicate and coordinate their work both through email and through the code itself.

Surrounding the core are perhaps ten times as many *co-developers*. These individuals contribute spo-

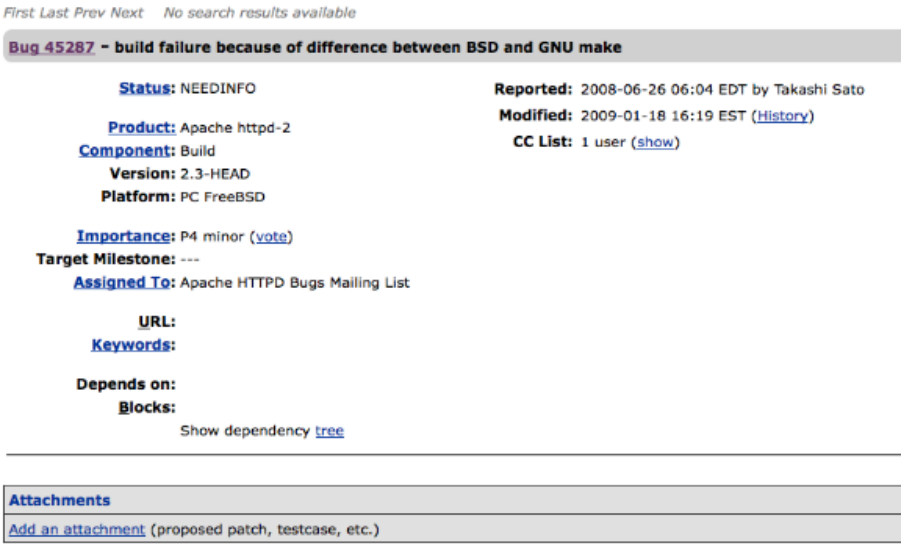


Figure 1. Example bug report from the Apache httpd project (from https://issues.apache.org/bugzilla/show_bug.cgi?id=45287).

radically by reviewing or modifying code or by contributing bug fixes. The co-developer group can be much larger than the core, because the required level of interaction is much lower. However, this lower level of interaction leads to the co-developers sharing less background knowledge than the developers.

Surrounding the developers are the *active users*: a subset of users who use the latest releases and contribute bug reports or feature requests (but not code). Some might argue that this last group should not be considered as part of the team, but they play an important part in the FLOSS development process. However, their interaction with the core is typically channelled through a constrained set of genres. For example, questions and bug reports from users are valued, but only if presented in the "right way" [25]. Since they are not otherwise involved in development, active users share even less background knowledge with developers.

As a source of examples to illustrate our theorizing, we examined documents and document use in three FLOSS projects, chosen to span a range of projects, with different sorts of developers and users: the Apache httpd project (http://httpd.apache.org/), as an example of a large institutionalized project with a mix of users, sophisticated as well as novice; MythTV (http://www.mythtv.org/), as an example of a medium-sized hobbyist-led project with a primarily consumer user base; and curl (http://curl.haxx.se/), a smaller project with an audience of more sophi–sticated users, since the product is a programming library and command line program. Including these three different kinds of projects allows us to determine if the usages we observe are common across a range of FLOSS projects or if there are differences related to the kinds of developers and users.

FLOSS projects create a variety of documents, including code, documentation, feature requests, bug reports and so on. To emphasize our theoretical comparison, we chose two kinds of documents with very different audiences, specifically; *bug reports* and source code control system (SCCS) *commit messages*.

An example bug report is shown in Figure 1. Bug reports are used to report problems with the system. They can be created by both end users and core developers, but are intended for core developers, since developers are the only ones who can actually fix bugs. However, a bug report can include discussions between users and developers, e.g., if developers request more information to characterize the bug. As a result, this kind of document often spans two distinct communities (users and developers) with little shared background. Each of the projects studied maintains a bug reporting

Figure 2. Example source code control system check in message from the MythTV project (from http://svn.mythtv.org/trac/changeset/24896).

cURL ▶ Docs ▶ **Bug Report**

How, Why, And Where to Report Bugs

Of course there are bugs in curl and libcurl. Some are known, but most of them remain unknown to us until you let us know. We depend on bug reports from the users to find the problems. We are not likely to be able to fix bugs if we don't get to know about their existance first. Bugs tend to get fixed within a short while after we've been notified (at least when we agree about it being a bug and not a feature)!

Known Bugs

Some bugs are known to already exist. We try to keep track of them on a separate KNOWN_BUGS document.

How To Report

If you can't fix a bug yourself, file an *as detailed a report as possible* in the bug tracking system **or** mail it to a suitable mailing list. Bugs in or improvements to libcurl are best posted to the curl-library list, while bugs in or with the curl tool can be posted to the curl-users list.

If you opt to use one of the mailing lists, notice that you need to be subscribed first to get the mail through to the list!

What To Report

When reporting a bug, try to include information that will help us understand what's wrong, what's expected to happen and how to repeat it. You should supply:

- your operating system's name and version number
- what version of curl you're using (curl -V is fine)
- what URL you were working with
- everything else you think might matter

Figure 3. Instructions for reporting a bug in curl (from http://curl.haxx.se/docs/bugs.html).

system with a page of explicit instructions about how and when to report a bug.

An example SCCS commit message is shown in Figure 2. A SCCS is a program that manages the source code for a project, keeping track of the current version and all revisions made to source files. A SCCS commit message is used to describe the revision made to a file when a new version is stored in ("committed to") the SCCS. These messages are created exclusively by and used primarily by core developers, meaning that this kind of document is typically shared amongst people with considerable shared background knowledge. Bug reporting systems can be made to interoperate with the source code control system, e.g., so that the commit message for changes that fix bugs can be linked to the bug report and vice versa.

4. Illustrative Examples

In this section, we provide illustrative examples of how the use of bug reports and SCCS commit messages in the three projects examined are consistent with the hypotheses developed above, taking each hypothesis in turn. We analyzed bug reports and commit messages for the three projects to determine to what degree the features specified by the hypotheses were present in each document type and across the two genres. In the current analysis, we used our judgement about the documents. A project for future research is to develop these intuitions into a more rigorous coding system for documents in order to more formally test our hypothesis.

Hypothesis 1: A document shared among people with asymmetric knowledge is more likely to explicitly state its purpose.

Examining the instructions for filing a bug report for the curl project (Figure 3) suggests that the purpose of bug reports is clearly stated: to let developers know about problems so they can fix them. The instruction pages for other projects are similarly explicit. By contrast, projects are less specific about the purpose of SCCS commit messages. While there are instruction pages for SCCS, e.g., as part of a description guidelines for the development process (e.g., Figure 4), they do not clearly state the purpose of the messages; rather, it seems to be taken for granted that the creator of such a message will know what information would be needed by other developers.

Hypothesis 2: A document shared among people with asymmetric knowledge is more likely to explicate the context of use:

A. By specifying the appropriate participants, times and places of its production and use

Consistent with this hypothesis, we find that bug report instructions seem somewhat more explicit about participants, time and places of production. In part, these expectations are enforced by the technology, as the bug tracking and SCCM systems enforce roles with particular privileges on documents, e.g., who can create, update, edit or dispose of certain kinds of documents. As well, some sets of instructions are very specific about what kinds of questions should be posted in which venues (e.g., Figure 3). Again, the instructions for the SCCM commit messages specify less.

B. Through ideal types, such as diagrams, atlas, road maps, which demarcate the specific elements or organization of the shared work.

Again, comparison of the instructions for the two types of documents seems consistent with the hypothesis. The instructions shown in Figures 3 list what the creator and receiver of a document have to do in order to demarcate the shared work. By contrast, these is little discussion of what someone might do when reading a SCCS message, perhaps reflecting an assumed shared understanding of the process.

C. By demarcating the boundaries of the shared work.

The instructions for bug reporting include descriptions of what is in scope and what is out of scope. For example, a complex system such as MythTV is built from many components. However, users rarely perceive these internal components of a system, and so consider all bugs as originating with the application they're use (e.g., see the note in the instructions about bugs in ffmpeg vs. bugs in nuvexport). The designers of a system may have specific use cases in mind, and may not be interested or willing to expand beyond those. Therefore, bug-reporting instructions need to explain how to localize a bug and caveats about what kinds of bugs can be fixed and what kinds of new features will be considered. In contrast, the description of the commit message does not specify such boundaries.

Hypothesis 3: A document shared among people with asymmetric knowledge is more likely to explicate the form and content of its communication by:

A. Bringing regularity in semantics and objects covered by one document to the next, e.g., through standardized forms that offer structured way to index communicative content.

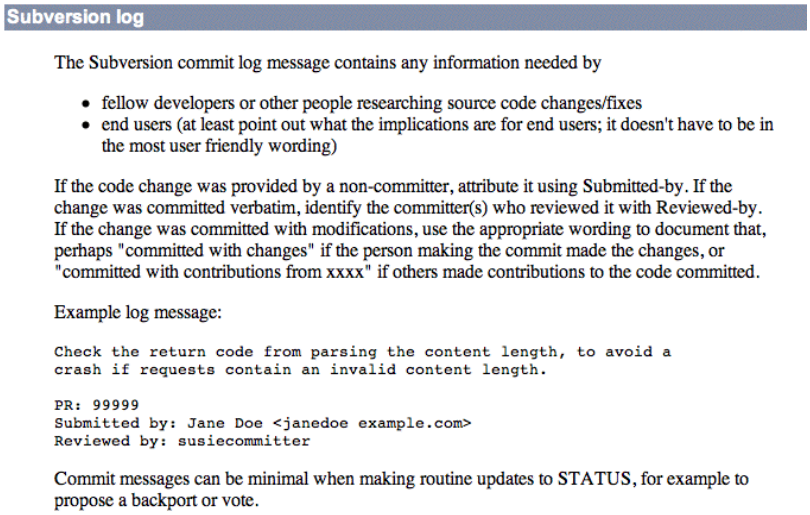
As expected, a bug report document includes a number of structured fields, e.g., as shown in Figure 1. The number of fields is greatest for the most institutionalized project, Apache, which uses the bugzilla bug tracking system. Interestingly, the curl project does not require a form but encourages submissions by email (as shown in Figure 3), asking only for some basic information. This difference may indicate the assumption that users of curl are sophisticated enough to submit good bug reports without explicit guidance. By contrast, a SCC commit message is just a plaintext field; the message include can be long or short. Some projects do suggest including particular fields, e.g., a reference to the bug report that the patch fixes, as shown in Figure 4, but these are not always required. Furthermore, exactly how the patch should be described is left to the developer.

B. Requiring users to make more details of their work visible in their descriptions.

The bug report document includes in addition to the fields describing the bug, comments made by developers or other users on the bug. These are frequently used to keep track of work status, as in the example in Figure 5 (next page), taken from MythTV. Commit messages are also used as a way to indicate and record the work done, though this is often done in only a summary fashion.

5. Discussion

Through our exploratory analysis of two kinds of documents, used across three FLOSS projects of various types, we found that documents supporting collaborators with asymmetric knowledge do seem to explicate their own use in more detail. Bug reports appear to do so by articulating or prescribing their own 1) purpose, 2) context of use, 3) content and form in greater detail than commit messages used by core community members with symmetric access to project knowledge. We will briefly discuss these findings in light of the literature.



Subversion log

The Subversion commit log message contains any information needed by

- fellow developers or other people researching source code changes/fixes
- end users (at least point out what the implications are for end users; it doesn't have to be in the most user friendly wording)

If the code change was provided by a non-committer, attribute it using Submitted-by. If the change was committed verbatim, identify the committer(s) who reviewed it with Reviewed-by. If the change was committed with modifications, use the appropriate wording to document that, perhaps "committed with changes" if the person making the commit made the changes, or "committed with contributions from xxxx" if others made contributions to the code committed.

Example log message:

```
Check the return code from parsing the content length, to avoid a
crash if requests contain an invalid content length.

PR: 99999
Submitted by: Jane Doe <janedoe example.com>
Reviewed by: susiecommitter
```

Commit messages can be minimal when making routine updates to STATUS, for example to propose a backport or vote.

Figure 4. Example instructions for SCCS commit messages (from http://httpd.apache.org/dev/guidelines.html).

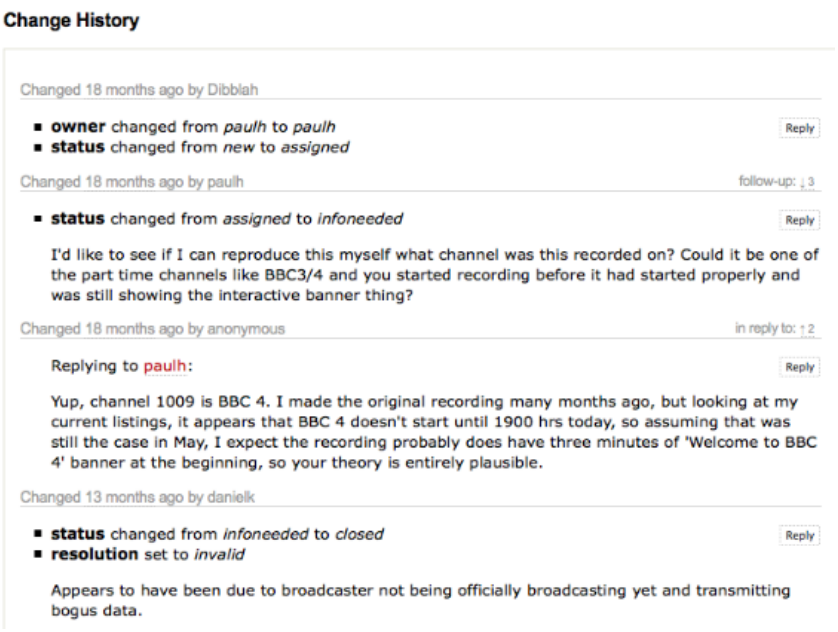


Figure 5. Comments on a bug report used to track work status (from http://svn.mythtv.org/trac/ticket/5943).

First, bug reports, which span two distinct communities with little shared background knowledge, in combination with explicit instructions, did articulate their own *purpose* more explicitly than commit messages used by core developers holding a considerable stock of background knowledge. This result raises an interesting theoretical contradiction. Genre theories argue that a shared purpose constitutes one of the defining characteristics of a document genre. In contrast, the literature on boundary objects emphasizes that one of the key benefits associated with these types of documents are their ability to support multiple purposes of different communities. Star highlights that different people can draw on boundary objects based on their individual purposes without having to negotiate differences in purpose [11]. One way out of this conundrum is to assume that boundary-spanning documents such as bug reports support different levels of purposes. They might come with an overarching purpose around which all users can rally. At the same time, each community can use the document for their own sub-purposes, which do not have to align. Future research may explore whether one can find such layering of purposes in document genres more generally.

Second, our exploratory analysis suggests that the bug reports do articulate their own *context of use* in greater detail than commit messages. Some of these prescriptions are built into the functionality of the systems such as who can participate, when and where. Others context-explicating devices, such as ideal types and boundary demarcation, are conveyed through textual means, figures and tables. These findings support prior research suggesting that documents can serve as portable places by demarcating the participants, timing and places of a document and the collaboration associated with their use [4, 26]. It is a portable place in the sense that the document explicates the context for a particular collaboration, thus assisting the participants in when, where and how they are expected to coordinate their activities and communication. This being said, more research is needed to fully understand how diagrams, maps and other visuals can support the communication among people with asymmetric access to knowledge. We have only scratched the surface when it comes to articulating how documents prescribe their own context of use. We can expect that organizational members engage multiple strategies to convey the context, e.g., through the use of security features or role-based functionalities.

Third, the *content and format* of bug reports prescribed more detail and regularity in semantics and objects covered compared to commit messages. It is not too surprising that documents linking people with little shared background knowledge explicate their content in much more detail. However, the use of standardized forms to explicate document use does bring another perspective to the debate around large structured information systems that may help us understand appropriate uses of standardized form when supporting groups varying access to knowledge.

5.1 Implications for research

The present paper offers a mere theoretical scaffold for exploring the characteristics of documents supporting work within and across organizational boundaries. Further conceptual work and literature review are required to fully develop such a perspective. The use of genre theory in this paper, for instance, suggests that it may be fruitful to explore how genres take shape and are shared in heterogeneous communities. Genre studies to date have tended to focus on groups with symmetric access to genre expectations. Future research could explore how genre expectations develop and get shared among people with asymmetric access to genre expectations. In short, how do genres work across various discourse community boundaries?

The literature on boundary objects and spanning has ballooned over the past decade. We only grazed the surface of this work. Further mining this source may bring up other valuable insights as to the characteristics of documents spanning collaborators with asymmetric access to knowledge. Other theoretical frameworks may also enlighten our endeavor. For instance, the work by linguistic anthropologist William Hanks suggests that one finds higher uses of indexical referential terms such as pronouns, demonstratives and deictics among people with symmetric access to knowledge [27]. Such communities can more freely use terms like "I, you, them, now, here, there, below, next" because their stock of knowledge help them decipher to what those terms refer in the shared context. Linguistic analysis of documents targeted for people with little symmetric access to knowledge versus documents used by groups with symmetric access to knowledge may prove fruitful.

Empirically, we hope to extend the present research beyond FLOSS teams, for example to online communities such as the Wikipedia community. Wikipedia does have an inner group that has intimate knowledge of the system and how the organization behind it works, and a larger peripheral group of participants with a much smaller stock of background knowledge. It would be interesting to explore why Wikipedia does not seem to require documents comparable to bug reports that bridge groups with asymmetric access to knowledge. Research could search for and describe other kinds of documents that bridge between these groups. It could be that there is no need to account for one's work in Wikipedia, as any member can commit a change to the core text. In contrast, only core developers can change the code in open source projects, thus requiring many would-be contributors to rely on communication with others to accomplish their work. Power relations and access to execute actions are thus likely to play a role in how much documents prescribe their use in various situations.

5.2 Implications for practice

The present research, even in its nascent state, offers some suggestions for information system design. In particular the extensive use of standardized forms in the bug report compared to the commit messages may provide some interesting insights. In healthcare, for instance, one finds a push for more standardized record keeping and information sharing. If mainly groups with asymmetric access to knowledge benefit from using standardized forms, one may assume that resistance to standardized systems mainly comes from groups with relative symmetric access to knowledge in their use of healthcare information systems. Using a standardized form that require high regularity in semantics and objects and great detail may simply seem like a waste of their time for someone with a large stock of background knowledge in the specific area. Large ERP systems may face the same issues. A detailed understanding of what characterize documents that support collaborators with symmetric versus asymmetric access to knowledge would help create systems that tailor content to specific user groups. The exploratory nature of the present study obviously calls for much more rigorous research in this area before one can harvest the benefits in practical system design.

6. Conclusion

Work in organizations, whether face-to-face or at a distance, bring together people with various access to and understanding of the work at hand. Nevertheless, most of them engage in documentation, whether as writers or readers. Yet, how do documents serve such diverse users, many of whom are literally not on the same page? The present paper has taken a first step towards building a framework articulating the characteristics of documents serving collaborators with asymmetric access to knowledge versus documents supporting people with symmetric knowledge. Drawing on document-centric approaches we hypothesized that documents supporting asymmetric groups are likely to be more prescriptive and explicate their own use compared to documents supporting symmetric groups. Through our exploratory analysis of two kinds of documents, used across three FLOSS projects of various types, we found that documents supporting collaborators with little shared background knowledge do seem to explicate their own use in more detail. They do so by articulating or prescribing their own 1) purpose, 2) context of use, 3) content and form in greater detail than documents used by core community members with symmetric access to project knowledge.

7. References

1. R. Harper, *Inside the IMF: An ethnography of documents, technology, and organizational action*, Academic Press, 1998.

2. D.E. Smith, *Texts, Facts, and Femininity: Exploring the relations of ruling*, Routledge, 1990.

3. M. Hartswood, et al., "Making a Case in Medical Work: Implications for the Electronic Medical Record," *Computer Supported Cooperative Work*, vol. 12, no. 3, 2003, pp. 241-266.

4. J.S. Brown and P. Duguid, "Borderline Issues," *Human-Computer Interaction*, vol. 9, no. 1, 1994, pp. 3-36.

5. C. Geertz, *The Intrepretation of Culture*, Basic Books, Inc, 1973.

6. D.E. Smith, *Institutional Ethnography: A sociology for people*, AltaMira Press, 2005.
7. C. Bazerman, "System of genres and the enactment of social intentions," *Genre and the New Rhetoric*, A. Freedman and P. Medway, eds., Taylor & Francis, 1995, pp. 79-104.
8. W.J. Orlikowski and J. Yates, "Genre repertoire: The structuring of communicative practices in organizations," *Administrative Science Quarterly*, vol. 39, no. 4, 1994, pp. 541-574.
9. K. Crowston and B.H. Kwasnik, "Can document-genre metadata improve information access to large digital collections?," *Library Trends*, vol. 52, no. 2, 2003, pp. 345-361.
10. J. Swales, *Genre Analysis*, Cambridge University Press, 1990.
11. S.L. Star and J.R. Griesemer, "Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology 1907-39," *Social Studies of Science*, vol. 19, 1989, pp. 387-420.
12. G.C. Bowker and S.L. Star, *Sorting Things Out: Classification and its consequences*, MIT Press, 1999.
13. E.A. von Hippel, "Innovation by user communities: Learning from open-source software," *Sloan Management Review*, vol. 42, no. 4, 2001, pp. 82–86.
14. E.A. von Hippel and G. von Krogh, "Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science," *Organization Science*, vol. 14, no. 2, 2003, pp. 209–213.
15. E.S. Raymond, "The cathedral and the bazaar," *First Monday*, vol. 3, no. 3, 1998.
16. P. Wayner, *Free For All*, HarperCollins, 2000.
17. B. Kogut and A. Metiu, "Open-source software development and distributed innovation," *Oxford Review of Economic Policy*, vol. 17, no. 2, 2001, pp. 248–264.
18. W. Scacchi, "Free/Open Source Software Development Practices in the Computer Game Community," *IEEE Software*, vol. 21, no. 1, 2004, pp. 56–66.
19. A. Cox, "Cathedrals, Bazaars and the Town Council," 1998; http://slashdot.org/features/98/10/13/1423253.shtml.
20. C. Gacek and B. Arief, "The many meanings of Open Source," *IEEE Software*, vol. 21, no. 1, 2004, pp. 34–40.
21. J.Y. Moon and L.S. Sproull, "Essence of distributed work: The case of Linux kernel," *First Monday*, vol. 5, no. 11, 2000.
22. M.A. Rossi, *Decoding the "Free/Open Source (F/OSS) Software Puzzle": A survey of theoretical and empirical contributions*, Working paper 424, Università degli Studi di Siena, Dipartimento Di Economia Politica, 2004.
23. J.M. González-Barahona and G. Robles, "Free Software Engineering: A Field to Explore," *Upgrade*, vol. 4, no. 4, 2003, pp. 49–54.
24. P. Giuri, et al., *Skills and openness of OSS projects: Implications for performance*, Working paper, Laboratory of Economics and Management, Sant'Anna School of Advanced Studies, 2004.
25. E.S. Raymond and R. Moen, "How to ask questions the smart way," 26/10/2009 2006; http://catb.org/~esr/faqs /smart-questions.html.
26. C. Østerlund, "Documents in Place: Demarcating Places for Collaboration in Healthcare Settings.," *Computer Supported Cooperative Work (CSCW)*, vol. 17, no. 2-3, 2008, pp. 195-225.
27. W.F. Hanks, *Intertexts: Writings on language, utterance, and context*, Rowman & Littlefield, 2000, p. v, 327.