A study of online discussions in an Open-Source Software Community: Reconstructing thematic coherence and argumentation from quotation practices

Flore Barcellini [(1)], Françoise Détienne [(1)], Jean-Marie Burkhardt [(1,2)], Warren Sack [(3)]

(1) INRIA-CNAM Eiffel Group, France

(2) Université Paris 5, France

(3) University of California, Santa Cruz, USA

Flore.Barcellini@inria.fr, *Francoise.Detienne@inria.fr*, *Jean-Marie.Burkhardt@univ-paris5.fr*, *wsack@ucsc.edu*

Abstract. This paper presents an analysis of online discussions in Open Source Software (OSS) design. The objective of our work is twofold. First, our research aims to understand and model the dynamics of OSS design that take place in mailing list exchanges. Second, our more long term objective is to develop tools to assist OSS developers to extract and reconstruct design relevant information from previous discussions. We show how quotation practices can be used to locate design relevant data in discussion archives. OSS developers use quotation as a mechanism to maintain the discursive context. To retrace thematic coherence in the online discussions of a major OSS project, Python, we follow how messages are linked through quotation practices. We compare our quotation-based analysis with a more conventional, thread-based analysis of the (reply-to) links between messages. The advantages of a quotation-based analysis over a thread-based analysis are outlined. Our approach provides a means to analyze argumentation and design rationales and promises a novel means to discover design relevant information in the archives of online discussions. Our analysis reveals also the links between the social structure and elements in the discussion space and how it shapes influence in the design process.

Introduction

This paper presents an analysis of online discussions in Open Source Software (OSS) design. In OSS design, the Internet plays a very important role (Raymond, in DiBona et al., 1999). As Mockus, Fielding and Herbsleb (2000) state: "co-designers work in arbitrary locations, rarely or never meet face-to-face and coordinate their activity almost exclusively by means of e-mail and bulletin boards. One consequence of this is that virtually all information of an OSS project is recorded in electronic form." OSS design is distributed and mostly asynchronous. It takes place in three activity spaces (Sack et al, 2004): (1) the implementation space constituted by code archives and the mechanisms of versioning systems (e.g., CVS); (2) the documentation space predominantly authored, stored and distributed as web pages; and, (3) the discussion space in which messages and comments are exchanged in newsgroups, mailing lists, weblogs, and chat environments.

Our research aims to understand and model the dynamics of OSS design that take place in mailing list exchanges; i.e., within a specific area of the discussion space. Our second, long-term objective is to develop tools to assist OSS developers in the extraction and reconstruction of design-relevant information from previous discussions. A large part of the OSS design process takes place in the discussion space and is archived in the documentation space. Developers new to an OSS project are encouraged to study what has already been tried and accomplished. Considering the huge quantity of data generated and archived, proposing methods and tools to extract relevant data, especially design rationales, from the design discussions addresses a real need.

We show how quotation practices can be used to locate design relevant data in online discussion archives. Until now the dominant model used to represent conversation has been a based on the "reply-to" links, the threading, between messages. Our approach is based on quotation rather than threading. We understand quotation to be a context-preserving mechanism used in online discussions (cf., Eklundh and MacDonald 1994). In synchronous, e.g., face-to-face, discussions, participants take "turns." Frequently, a turn is a reply to the previous turn. For example, when one participant raises a question, in the next turn someone might answer the question. Thus, conversation analysis frequently entails finding adjacency pairs like question-answer or greeting-greeting, etc.. Within newsgroup or email-based discussions, quotation supports adjacency by maintaining two turns within a single message. In other words, by quoting the text of the previous message, one's message can incorporate both a question and an answer, or any number of other such adjacency pairs.

Our working hypothesis is that quotation-based representations are better than threading-based representations for the reconstruction of thematic coherence and for identifying and highlighting design activity that takes place within online

discussions. We also hypothesize that quotation practices are linked to the social structure of an OSS project, specifically to the roles and differences of influence performed by project participants.

From among a wide variety of ongoing Open Source Software (OSS) projects, we have chosen to investigate the design processes of a major OSS project devoted to the development of a programming language called Python.

In the following sections we first review some prior studies of software design activities and the role of argumentation in the articulation and communication of design rationales. We then review prior work in thematic coherence analysis and in the analysis of quotation in online discussions. Finally, we present our quotation-based methodology and discuss our results.

Argumentation, collaboration and software design

Many previous studies of software design have analysed collaborative activities that take place in face-to-face meetings; e.g., brainstorming and technical review activities (D'Astous et al, 2001; 2004; Herbsleb et al. 1995; Olson et al. 1992; Walz et al. 1987). Researchers have identified various types of collaborative design activities.

One set of collaborative activities is related to the objects of design. These activities concern the evolution of the design problem and solution; e.g., elaboration of the problem and the enhancement or identification of alternative solutions. Evaluative activities – e.g., the evaluation of solutions or the articulation of alternative solutions – are also of this kind.

Another type of activity concerns the construction of common references, or common ground, by a group of co-designers. For example, clarification or cognitive synchronization activities take place when a group negotiates or constructs a shared representation of the current state of the solution.

Group management activities are a third kind of design activity. These activities are frequently related to issues of process. Project management activities that concern the coordination of people and resources - e.g., the allocation and planning of tasks – are of this kind. Meeting management activities – e.g., the ordering and postponing of topics of discussion – are another example of this kind of activity.

Co-designers accomplish design and evaluation activities by arguing with each other. These arguments have a very specific form and can be characterized as a sequence of "moves" or "turns."

For example, D'Astous et al. (2004) analyzed the argumentative moves in software technical review meetings. They found, for instance, that the elaboration of a solution can be followed immediately by either its evaluation alone or its evaluation and development of an alternative solution. Such review activities may, or may not, be preceded by a cognitive synchronization exchange. A

cognitive synchronization exchange allows designers to articulate a shared representation of a design before it is evaluated. This argumentative move is referred to as a proposition-opinion. The review of a solution, in particular a negative review, leads participants to develop alternative solutions. An alternative solution may be a justification for the negative review or an answer to the weaknesses identified in the negative review. D'Astous et al. called this move opinion-arguments.

Argumentation makes explicit the design rationale; i.e., the reasoning behind the design of an artefact. By making their rationales explicit, designers have the means to keep track of past decisions and communicate these rationales to others outside the design team (Buckingham Shum and Hammond, 1994; Concklin and Burgess, 1991; Moran and Carroll, 1996). Different methodologies have been proposed to keep track of design rationales in design meetings. Unfortunately, designers often see these methodologies as imposing extra work on them; work that does not yield any immediate benefits for them. Our long term aim is to avoiding this objection by building tools capable of automatically extracting design-related information from archives of online discussions.

Open source software design

Open-source software design is a particular case of asynchronous, distributed, collaborative design (DCD). Descriptions of OSS design (DiBona et al, 1999; Elliott and Scacchi, 2004; Mockus et al. 2000; Raymond, 1999; Stallman, 2002) often highlight the following points:

- OSS systems are frequently built by large numbers of volunteers;
- work is not assigned; people undertake the work they choose;
- there is no explicit, system-level design;
- there is no project plan, schedule, or list of deliverables;
- work is done almost exclusively at a distance.

Empirical studies of the social organization and the dynamics of design processes of specific OSS projects have shown that these points constitute an idealized picture of OSS design. Specific OSS projects diverge from the idealized picture in a number of different ways.

For example, some OSS communities have a strict, hierarchical organization that stratifies developers into levels (Gacek and Arief, 2004; Mahendran, 2002). Centralized power structures of this sort are at odds with the flat, merit-based structure idealized by many OSS communities. The community we focus on, Python, has a very centralized organization. See Mahendran (2002) for an ethnographic study of the Python project and description of its hiearchicalized organization. The Python core developers (referred to as "administrators") have more power than ordinary co-developers in making executive decisions and modifying the code.

OSS design processes are not always as open-ended as the idealization might imply. Certain projects have prescribed means for controlling task assignment and for setting project plans and schedules. For example, the designers of Python engage in a specific design process called Python Enhancement Proposals (PEPs). PEPs are the main means for proposing new features, for collecting community input on an issue, and for documenting chosen design decisions. Some PEP documents describe new features of Python. Others specify more general information about the processes or organization of the Python community. When a PEP is written to describe a new language feature, it is suppose to provide a concise technical specification of the feature, a rationale for the feature, and a reference implementation.

The process of writing, reviewing and implementing PEPs is quite similar to two design processes used in conventional software projects: Request For Comments (RFCs) and technical review meetings. RFCs have been practiced for decades to define standards for the Internet (especially by the Internet Engineering Task Force, IETF). Technical Review Meetings (D'Astous et al., 2004) have been practiced in many corporate and governmental settings.

In Sack et al (2004) we have analyzed the PEP design process as a set of activities that take place in three different spaces: the discussion space, the documentation space, and the implementation space. Figure 1 shows an overview of the PEP process with links to these three activity spaces. Once a rough-draft PEP is accepted, the author of the PEP, called the champion, is responsible for posting the PEP to the community forums where the PEP is discussed. Archives of discussion, decisions regarding the PEP, and the different versions of a PEP are kept in the documentation space. Information about and the status of a PEP is, therefore, distributed between these two spaces. After a PEP has been accepted, it is given a final review by the leader of the Python project. Finally, if a consensus reached, a new piece of code is written to implement the PEP. This code is integrated into the project's code archive: the implementation space.

Previous studies of OSS design projects have focused on different activity spaces. Mahendran's (2002) ethnographic work illustrates how power is distributed across the three activity spaces - the discussion, implementation and documentation spaces. Ducheneaut's (2003) work investigated the evolution of links between people in two activity spaces – the discussion and implementation spaces – and showed how newcomers can be (but sometimes are not) progressively integrated into the social and the technical structure of the Python project. Sandusky et al. (2004) focused their analysis on the documentation space of the Bugzilla project. Mockus et al. (2000) focused their analysis on the implementation space. In this paper, we examine the dynamics of the discussion space and examine the influence of the social structure of the Python project on the discussion space.

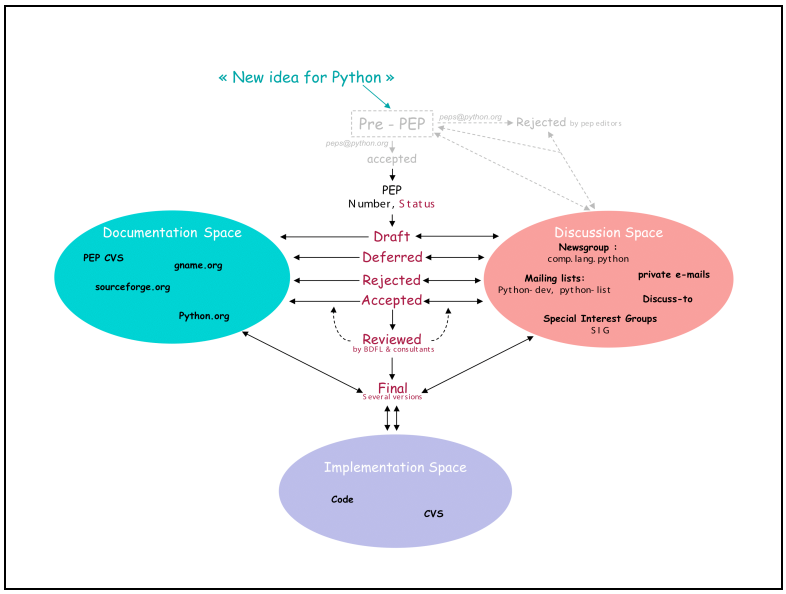

Figure 1: Overview of the Python Enhancement Proposal (PEP) process

Thematic coherence and quotation practices in online discussions

A large part of OSS design takes place in a discussion space where messages are exchanged between participants. Thematic coherence concerns how a message connect to previous messages of an exchange. In face-to-face conversation, coherence concerns how a turn connects to previous turns in a dialogue. Coherence in face-to-face conversation can be seen as actively constructed by participants across turn taking. In contrast to the face-to-face situation, in online conversations, a message can be separated both in time and place from the message it responds to. So, some form of explicit (or inferable) link between messages is usually required to understand the thematic coherence of an online discussion.

Current work on online discussions (e.g., Venolia and Neustaedter, 2003; Popolov et al. 2000) frequently assumes that the conversational structure is determined by "threading." I.e., the assumption is that the thematic coherence is determined by the "reply-to" links established between messages when participants reply to already posted messages. When a participant composes a reply to another message, the message-ID of the original message is placed in an "In reply-to" header on the reply message. The use of these message-to-message reply links and the outline-style presentation of message threads is practically universal among all known popular email software.

The threading approach is the main basis of tools for organizing and visualizing online discussions; e.g., Threaded Chat (Smith et al., 2000); Chat

Circles, Loom (Donath et al., 1999); and, Netscan (Smith et al., 2001). These tools suffer from several limitations. Some of them work with existing e-mail or newsgroup discussions, hence requiring no change in practice, but yield relatively little conversational structure (only basic threading). Others build rich conversational structures, but require a large change in practice: participants cannot use their usual desktop applications to post and read messages; they need, instead, to use the prototype tools developed by researchers.

The threading model is useful for analyzing conversational roles and for mapping the centrality of participants in a social network (see, for instance, Viégas and Smith, 2004). However, the threading model has some important limitations. Herring (1999) outlines how, in the threading model of online conversations, turn adjacency is disrupted; i.e., relevant responses do not occur temporally adjacent to initiating turns; e.g., an answer to a question might not arrive until long after the question is posted. This is a violation of sequential coherence (pragmatic principles of adjacency and relevance). Thus, this model provides an overview of the conversation but it is cannot correctly characterize its referential coherence.

To avoid this limitation, we propose to use quotations as the links to extract coherence in online conversations. Eklundh and Macdonald (1994) showed that quoting a message -- i.e., including it in a comment or reply -- was a widely used technique in e-mail dialogues. Quoting is seen as a context-preserving mechanism but the majority of responders use it selectively. Eklundh and Macdonald results showed that conversational participants perceived the use of quoting as contributing to the sense of the conversation when communicating in e-mail. Quoting is seen as a linguistic strategy (Eklundh and Rodriguez, 2004) used by participants to connect a comment to previous contributions to the conversation. Quoting creates the functionality of adjacency (Herring, 1999): it incorporates portions of two turns within a single message. It maintains context (i.e., portions of previous messages) and so can be use to retrace the history of a conversation.

As far as we know, there have been only two attempts to develop tools to automatically identify quotations and to represent online conversations based on quotation links between messages: Conversation Map (Sack, 2000) and a prototype inspired by Conversation Map called Zest (Yee, 2002). Our study expands on this work by analyzing quotation practices and participants' conversation roles within the context of a design activity, the design of OSS.

Study of online discussions in the Python OSS project

Corpus

Our message corpus was drawn from one of the major elements of the discussion space of the Python community: the python-dev mailing list hosts discussions pertinent to design decisions. We selected one conversation regarding a specific PEP (PEP 279). The corpus contains a total of about 3800 lines of text. The entire conversation is archived on the web and is public

PEP 279 proposes three different enhancements to Python: (1) a new index function; (2) a way to facilitate generator comprehension; and, (3) a means for generator exception passing. The corpus analyzed is composed of two discussions: part one (73 messages posted by 21 authors between March 28th and April 8th 2002) and part two (58 messages posted by 29 authors between April 24th and April 27th 2002).

Method

Our objective is to determine if correlations can be found between a participant's status and the patterns of quotation employed in a participant's posted messages. Our method is structured around the analysis of three aspects of online discussions:

(1) quotation practices and message structure;
(2) characterization of participation within the discussions and the declared status of participants in the project;
(3) message content and activities analysis.

In the following, each message will be characterized according to these three aspects.

Quotation practices and message structure

We have observed that quoting is a general strategy employed by participants in the PEP 279 discussion:

- 84% of messages in part one of the corpus and 90% in part two contained at least one quote;
- 24% of message lines in part one and 34% of the lines in messages from part two are quoted lines;
- Half of the authors accounted for at least 16% (median) of the lines quoted.

Looking at the messages and the ways they were (or were not) quoted, we observed a similar set of results in both part one and part two of the corpus:

- 41% (in part one) and 47% (in part two) were not quoted at all;
- 29% (in part one) and 19% (in part two) were quoted once;

- 30% (in part one) and 35% (in part two) were quoted by between two and six different messages.

We categorized messages according to the alternation of blocks of quoted material and blocks of commentary (new text) in a message:

- A text-only message (TO), is a message that does not contain any quotations;
- A one-quote message (1q) is a message with one block of quotations followed by a comment. We distinguish two kinds of 1q messages:
  - One quote-one source messages (1q-1s): these messages contain one quotation from one source message;
  - One quote-multiple source messages (1q-Ms): these messages contain one block of quotations, but the quotations includes text from two or more source messages followed by one block of commentary;
- A multiple-quotes message is a message containing alternating quotes and comments (Mq). We distinguish three kinds of Mq message:
  - Multiple quotes-one source (Mq-1s) messages: several quotes of the same source message;
  - Multiple quotes-multiple sources (Mq-Ms) messages: embedding of quotations from several source messages;
  - Multiple quotes-multiple sources "composed" (Mq-MsC) messages: composition of quotations from several source messages.

Using these definitions, each message is categorized according to its structure and the source message(s) that is (are) quoted by the message. Aggregating part one and part of our corpus of messages we found that message structures are distributed as following:

- 9% text-only messages;
- 70% one-quote messages;
- 21% multiple-quotes messages.

Comparing discussion participation with participants' declared status in the Python project

Two major variables that might affect quotation practices include the level of participation exhibited by project members within the discussion list (python-dev) and a member's declared status within the Python project (as declared outside of the discussion list; e.g., the project administrators are declared on the project website: http://sourceforge.net/projects/python/).

As Mahendran (2002) pointed out the Python project has a centralized social structure. One can identify four important, declared roles:

- The project leader sometimes referred to (semi)-ironically as the BDFL (Benevolent Dictator For Life);
- The champion of the PEP: the one who proposes and writes the PEP. In our example discussion (concerning PEP 279) the champion is a project developer.

- The core team or administrators: nine people (at the time of our analysis) who are co-located with the project leader in a corporation called Zope. Their role is to maintain the code base, the documentation, and the PEP process.
- The developers: Only the project leader can accept a new developer into the list. To be accepted, new developers need to have demonstrated proficiency in Python. They are geographically distributed throughout the world.

To distinguish levels of participation in the online discussion, we have divided the population into two groups according to the median number of messages posted:

- HP-A/Dev: Administrators (including the project leader) and developers (including the champion) who sent more than two messages are High Participant Administrators (HP-A) or High Participant Developers (HP-Dev);
- LP-A/Dev: Those who posted fewer than two messages are termed Low Participant Administrators (LP-A) or Low-Participant Developers (LP-Dev).

Message content and activity analysis

Our message content analysis is a more fine-grain analysis based on a method developed in the field of cognitive ergonomics of design. Blocks of quotation or commentary contained in a message are categorized according to a coding scheme developed in our previous work (D'Astous et al. 2004; Détienne et al, 2003; Détienne et al. in press).

We identified the themes addressed by messages and found five themes corresponding to technical design problems:

(1) P1: this theme concerns the issue of what functions, to be built into the Python language, are to be named; twenty-three alternative names were proposed;

(2) P2: different possible syntaxes for the functions were discussed; eight such syntactic alternatives were articulated by the discussants;

(3) P3: concerned the syntax, semantics and history of a technical issue concerning generator comprehension;

(4) P4: concerned the technical issue of generator exception passing;

(5) P5: concerned an orthogonal problem of name binding and the status of name spaces (i.e., two other technical issues).

We also characterized the message content with respect to the following categories of design activity (or the rhetorical function of the message):

- proposal of an (alternative) solution;
- evaluation: agreement/disagreement;
- group coordination;
- synthesis;

- clarification;
- explicit decision;
- other activities.

These categories were used to label the quotations and the comments in the messages. The analysis was done manually by the first co-author of this paper and validated iteratively with the second and third co-authors.

Results

Quotation practices, message structure, and thematic coherence

Our analysis of quotation practices allows us to compare a representation of online discussion based on quotation-based links between messages with a representation based on threading or "reply-to" links between messages. Figures 2a and 2b illustrate these two different ways of representing the PEP 279 discussion. In the figures, the circles represent email messages (labelled with an arbitrary number). Arrows joining the circles symbolize either a "is-a-reply-to" or a "is-quoted-by" link between two messages. The circles are colored to represent the different themes (i.e., the different design problems, P1-P5, enumerated above) addressed by the messages.

Figure 2a is an analysis of the discussion based on the threading, or reply-to, links between messages. Using the reply-to links to partition the messages, it appears to be the case that the conversation is fragmented into three different threads. This analysis by threads also corresponds to the way in which the discussion is archived on the web (at the URLs cited above).

Figure 2b, an analysis of the discussion based on quotation-based links between messages, reveals a distinctly different organization of the messages. In this analysis all of the messages are connected together, rather than the three distinct threads shown in figure 2a. In this analysis almost every message is linked to another message and the thematic coherence of the discussion is preserved. There are only three text-only messages that needed to be linked to the others using a reply-to relationship.

In Figure 2b, four areas can be discerned: at the beginning of the conversation, the four themes (P1, P2, P3 and P4) are treated simultaneously in the messages (black circle) except for two messages that discuss only P2. Immediately thereafter two themes, P1 (blue circles) and P4 (pink circles), are the foci of discussion. Finally, an orthogonal problem, P5, emerges (orange circles).

The thematic coherence of the discussion, especially regarding P1, is better represented by the quotation-based links of Figure 2b than by the reply-to links of Figure 2a. Moreover, closer examination of the message contents reveals that the messages that are unlinked in Figure 2a are pivotal to the overall discussion. For example, message 68 initiates a discussion and constitutes a set of "opening remarks" crucial to the rest of the discussion. Message 4 generated several

diverging branches of discussion. By comparing the position of messages 4 and 68 in figure 2a with their positions in figure 2b, one can see that the reply-to representation does a poor job of positioning them where they should be. Figure 2a shows messages 4 and 68 in detached and peripheral positions. In contrast, Figure 2b, constructed from the quotation-based links between messages, positions them as they should be, namely, in the "thick" of discussion. These results are consistent with our working hypothesis that a quotation-based representation is better than threading for reconstructing the thematic coherence of design-related online discussions.

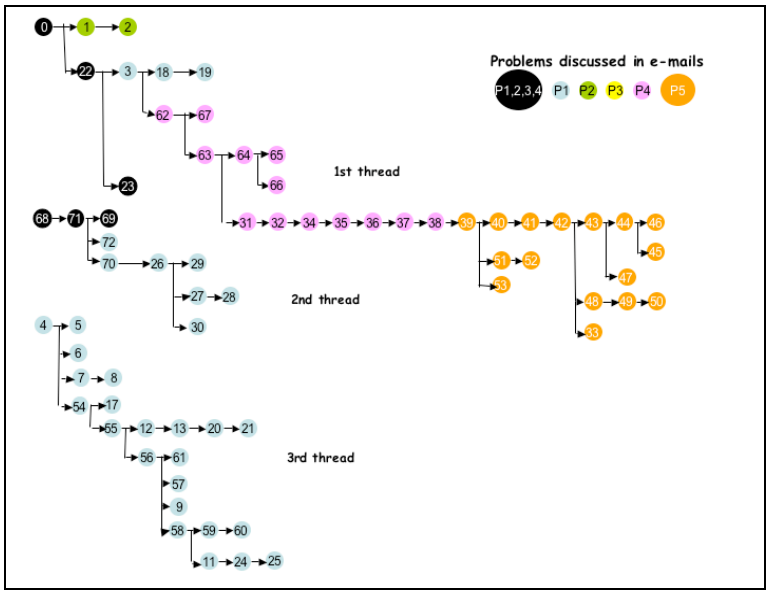

Figure 2a: Threading based representation of the links between messages

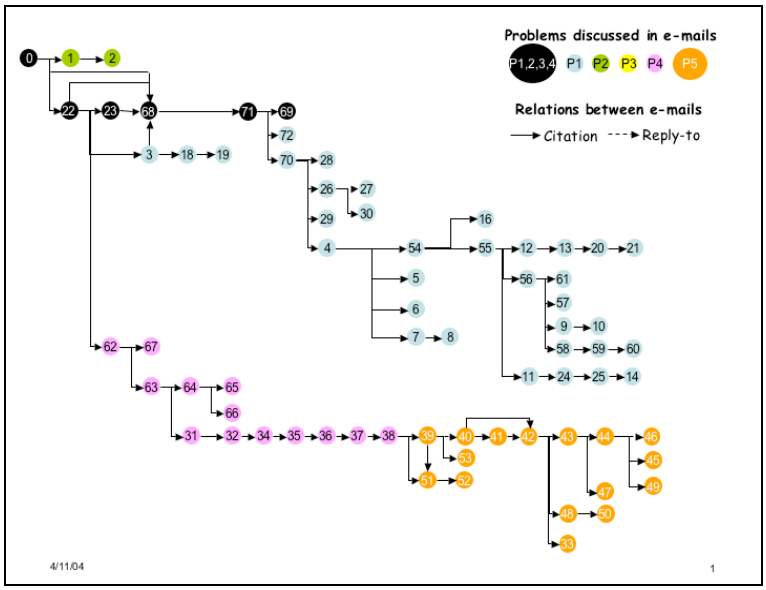

Figure 2b: Quotation-based representation of the links between messages

Finally, comparison of figures 2a and 2b shows that the set of messages that reply to a particular source message is a proper subset of the set of messages that quote the source (PEP 279 discussion). This suggests that quotation-based links contain more information that reply-to links.

In figure 3 we have layered another set of annotations on top of the graph shown in figure 2b (i.e., the graph constructed from quotation-based links between messages). The additional annotations in figure 3 outline groups of messages (with a dotted line) that all contain quotations from a given message. Figure 3 illustrates, what we will call, the "depth of quotation." Quotations of depth 1 are contained in messages immediately linked to the quoted message. Quotations of depth 2 are contained in messages that are linked to messages with quotations of depth 1; etc..

Figure 3 shows that the average depth of quotation is rather small. This result suggests that sub-thematic coherence could be constructed by partitioning messages into groups as was done for figure 3. More analyses would need to be done to determine if these message subsets based on the quotation of the same source correspond to a sub-thematic organisation.

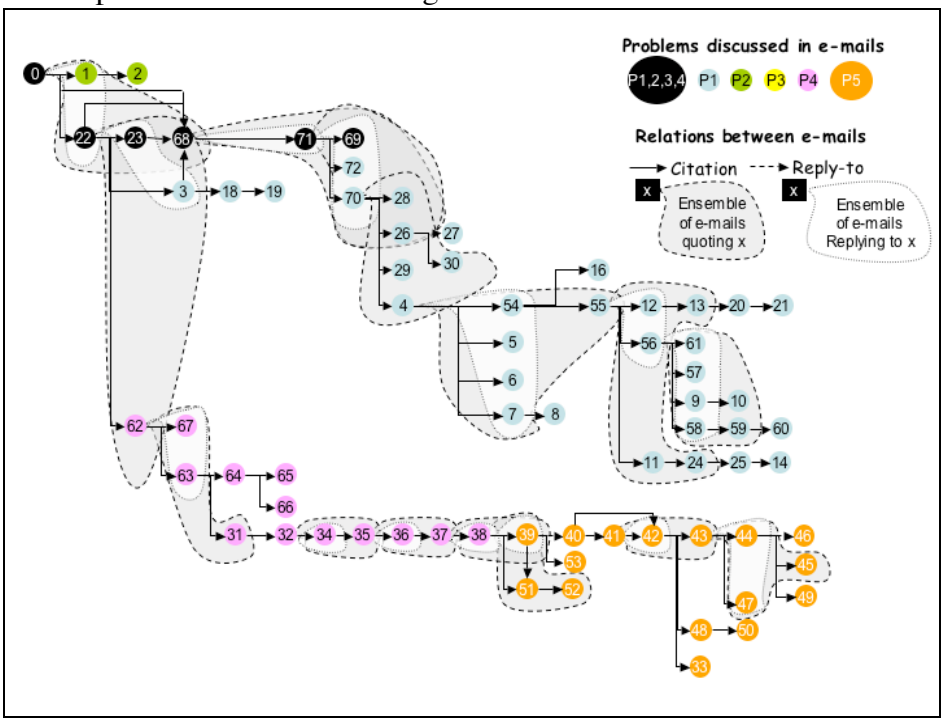

Figure 3: Sets of messages that quote particular source-messages and sets of messages that reply to the same source-messages

Quotation practices and degree of synchronicity

We also analyzed the flow of messages according to their posting time and the posting time of the messages in which they were quoted. Our objective was to

obtain an overview of the degree of synchronicity of the PEP discussion. The geographically-distributed nature of the project makes this an important issue to study. The results are as follows:

- 50% of the messages quoted were quoted for the first time within an hour following their posting; 75% were quoted within five hours;
- 50% of the messages quoted a second time were quoted a second time within an hour following their posting; 75% of the second quotations occurred within seven hours of the message's posting;
- 50% of the third and 50% of the fourth quotations occurred within twenty-four hours of the posting of the message; 75% within 48 hours.

According to these results, it seems that there is a large degree of synchronicity; or, stated otherwise, sub-discussions organised around the same design topics have a weak degree of asynchronicity. In fact, late citations are often posted by co-designers who are far away from the USA (where most participants are) and their messages then arrive after design decisions have been taken.

Discussion participation and assigned roles in the Python project

Figure 4 represents the same discussion but messages are labelled with the project roles of their posters. The figure shows that the patterns of quotation -- sequential versus branch structure -- tend to correspond with the social position of the poster in the Python project: (1) a branching structure (when multiple messages quote from a single message) is generally initiated by a message posted by either the project leader or the PEP's champion; (2) High-participant Administrators are usually the ones to post messages that close a line of discussion; (3) sequential structures tend to alternate between messages posted by administrators and messages posted by developers. However, in the thematic drift away into P5 this is not observed. Here, the project leader and the PEP's champion stop participating until, finally, the project leader ends the discussion (with message 50). This analysis shows a relationship between the social structure of the Python project and participation in the online discussion. The social structure influences the design process as it unfolds in the discussion space.

Figure 5 shows that the depth of quotation achieved by a message is related to the message poster's status in the project. The project leader and the champion do not only initiate branching structures; their messages are also quoted much more deeply (i.e., repeatedly in subsequent messages) than the messages posted by the other participants. These results are consistent with the fact that this project has a very centralized social organisation and they show that key participants have a greater influence than others on the conversation.

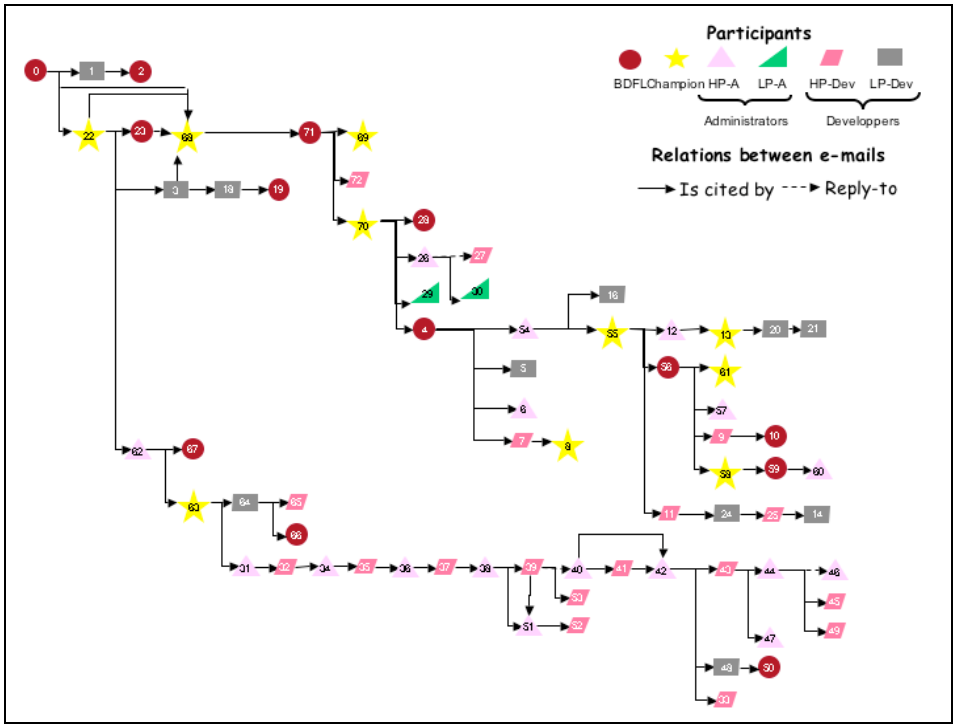

Figure 4: Status and position in the discussion

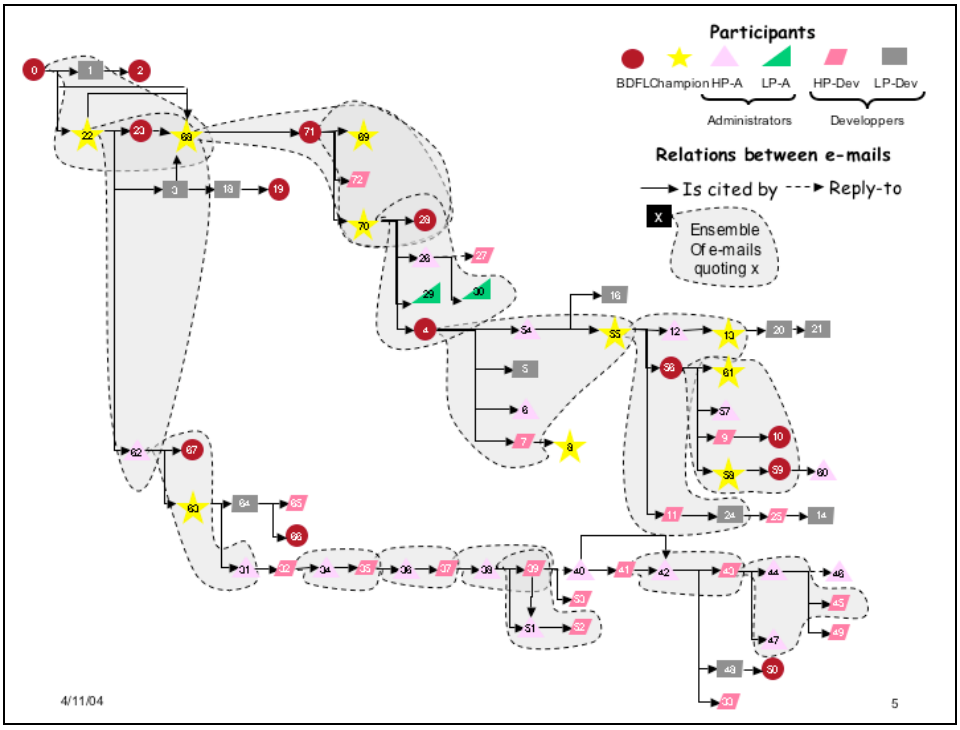

Figure 5: Depth of quotation and status of participant

Message content and design activity analysis

Analyzing the content of the quotations, we found that the most prevalent design activities quoted are Syntheses (N=49; i.e., 28%) and Disagreements (N=48; i.e., 28%) followed by Proposals (N=31; i.e., 18%) and Agreements (N=20; i.e., 12%). The PEP champion was the source of the largest number of quotations (N=40; i.e., 23%) followed closely by HP-Devs (N=37; i.e., 21%), HP-As (N=37; i.e.. 21%) and the project leader (N=36; i.e., 21%). Unsurprisingly, we found a smaller number of quotations from LP-Dev participants (N=20; i.e., 12%) and LP-As (N=3; i.e., 2%).

Interestingly, there is a relationship of intermediate strength between the status of the quoting participant and the nature of activity contained in the quotation (V2 Cramer = 0.07). V2 Cramer is an indication of the strength of the relationship between two nominal variables. The relationship is considered to be weak when V2 < 0.4; intermediate when 0.4 < V2 < 0.16; and, strong when V2 > 0.16. Our statistical analysis indicates the following strong links between the status of the participants and the content they quoted:

- The PEP champion mostly quoted Decisions; other activity; and, to a slightly lesser degree, Agreements. Conversely, he did not quote Syntheses. This is easily explained since he was the author of most of the synthesizing messages and he does not quote himself.
- The project leader quotes Syntheses and Proposals. Conversely, he tends not to quote Agreements or Disagreements.
- HP-As mostly quoted Disagreements, Agreements, and Coordination messages. Conversely, they tend not to quote Proposals.
- LP-As quoted Proposals and tended not to quote Disagreements.
- HP-Devs mostly quoted Clarifications and Disagreements. They tended not to quote other activities and Agreements.

We also categorized the subsequent comments according to design activity. The most frequent activities that appeared in comments were Agreements (N=66) and Disagreements (N=58). Hence, 57% of the comments correspond to an evaluation activity, meaning that evaluation is the main activity related to the usage of quotation. We found lower frequencies for activities such as Clarification (N=28; i.e., 13%) and Proposals (N=26; i.e., 12%). Finally, we observed very few Decisions (N=14; i.e., 6%), Syntheses (N=11; i.e., 5%), or comments of Coordination (N=6; i.e., 3%).

Additionally, we investigated the correlation between the type of comment posted and the status of its author. Overall, about half of the comments were authored by HP-As (N=103; i.e., 47%). The remaining comments were mostly distributed between the project leader (N=37; i.e., 17%), the PEP champion (N=32; i.e., 15%) and HP-Devs (N=29; i.e., 13%). Very few comments were due to LP-Devs (N=15; i.e., 7%) or LP-As (N=2; i.e., 1%). Descriptive statistics show that, globally, there is a weak relationship between the design activity of a

comment and the status of the participant (V2 = 0.03). Therefore, in general, the various types of activity are, roughly, equally distributed across all types of participants. However, it is remarkable that only the project leader posts Decisions.

We also analyzed the relationship between the set of activities quoted by a participant and the participant's subsequent commenting activity. Note that, in some cases, comments are preceded by a quotation that incorporates more than one design activity. Consequently, to perform the following analysis, we added the category "multiple activities" to our list of activity categories. We found a strong relationship between the type of activity in the quote and the nature of the activity in the associated comment (V2 = 0.23). In particular, we observed the following strong associations between activities contained in the quotation and in the subsequent comment:

- An Agreement is usually followed by a Proposal;
- A Disagreement is usually followed by a comment containing a Disagreement. Conversely, Disagreements are not usually followed by Agreements;
- An Agreement is usually followed by a Synthesis;
- A quotation of a Coordination comment is usually followed a Coordination comment;
- Clarifications are usually followed by Syntheses or a previous clarification quotation;
- Other activities are mostly associated with quotations presenting other activities.

Some of these associations may be interpreted in terms of argumentative moves as in D'Astous et al. (2004). Agreement-Proposal can be interpreted as an implicit disagreement justified by an alternative proposal. Conversely Agreement-Synthesis can be interpreted as an implicit agreement with a reinforcement of the consensus by a synthesis activity. Disagreement patterns (Disagreement-Disagreement) display diverging moves among participants. Finally, some patterns show an ongoing discussion of coordination (Coordination-Coordination) or are indicative of a co-construction of common knowledge (Clarification-Clarification and Clarification-Synthesis).

Furthermore, we found strong associations involving the Decision activity:

- A Proposal or Synthesis is usually followed by a comment containing a Decision;
- A Decision is usually followed by either a Proposal or a Coordination comment. Decision-Proposal pairs can be explained by the fact that some proposals are posted by geographically distant participants after a decision has already been made. Note that the strength of this result is probably exagerated by the low number of explicit decisions in the corpus. Decision-

Coordination pairs are apparent when a decision is made and then corresponding tasks are allocated to particular participants.

Discussion

Our study shows that a quotation-based analysis is a promising approach for identifying thematic coherence and design-relevant information in the archives of online discussions. A quotation-based analysis of thematic coherence was shown to be better than a thread-based analysis. The thread-based analysis incorrectly divided some theme-related messages into different threads and, furthermore, categorized as peripheral certain messages that were central contributions to the discussion. A quotation-based analysis did not exhibit these weaknesses.

Our content analysis of the messages revealed several interesting relationships between quotations and the comments that follow the quotations. We found that quotations are largely correlated with evaluative design activities. The relationships and correlations we have uncovered, between quotations and commentary, should aid us in the development of tools for archiving and visualizing online discussions. We intend to build on the quotation analysis procedures currently incorporated in the Conversation Map system (Sack, 2000) and, thereby, to provide some automated means to foster knowledge sharing in distributed collective practices.

Our analysis also revealed links between the organized social structure of the Python project and the shape of the discussion space. A participant's assigned role in the project organization affected whom the participant responded to in the online discussion and, therefore, influenced the unfolding of the design process within the discussion space. Two participants led the discussion we studied: the project leader and the champion of the PEP. This OSS community closely resembled the hierarchical organization of more traditional software design projects. This result can be opposed to the idealistic vision of OSS design.

Our study is an analysis of only one PEP discussion. PEP discussions can vary according to the status of the champion, according to whether the PEP has been accepted or rejected, and according to their (loose versus tight) coupling with other Python design tasks (Olson and Olson, 2000). In future work, we plan to replicate the analysis on a variety of other PEP discussions.

In order to further extend our analysis to a wider sample of corpora, we plan to automate some parts of the structure and content processing. Currently under development is software to automatically identify quotation links between messages. We also hope to construct software to automatically analyse themes of discussion computing (Sack, 2000); and, to analyze patterns of argumentation, an admittedly much more difficult task akin to rhetorical structure parsing (Marcu, 1997).

Acknowledgments

This study was supported by the France-Berkeley Fund; the French TCAN-CNRS program; and, the National Science Foundation, Directorate for Computer and Information Science and Engineering, Division of Information and Intelligent Systems, Digital Society and Technologies Program, Award 0416353.

References

Buckingham Shum, S., and Hammond, N. (1994) Argumentation-based design rationale: what use at what cost? International Journal of Human-Computer Studies, 40, 603-652.

Concklin, E. J., and Burgess, K. C. (1991) A Process-Oriented Approach to Design Rationale. Human-computer Interaction, 6, 357-391.

D'Astous, P., Détienne, F., Robillard, P. N., and Visser, W. (2001) Quantitative measurements of the influence of participants roles during peer review meetings. Empirical Software Engineering, 6, 143-159.

D'Astous, P., Détienne, F., Visser, W., and Robillard, P. N. (2004) Changing our view on design evaluation meetings methodology: a study of software technical evaluation meetings. Design Studies, 25, 625-655.

Détienne, F., Burkhardt, J-M., and Visser, W. (2003) Cognitive effort in collective software design: methodological perspectives in cognitive ergonomics. Proceedings of the 2nd Workshop in the Workshop Series on Empirical Software Engineering "The Future of Empirical Studies in Software Engineering", pages 17-25, Monte Porzio Catone (Rome, Italy), 29 September, 2003.

Détienne, F., Martin, G., and Lavigne, E. (in press) Viewpoints in co-design: a field study in concurrent engineering. Design Studies.

DiBona, C., Ockman, S. and Stone, M. (1999), Open Sources: Voices from the Open Source Revolution, O'Reilly and Associates Inc., Sebastol, CA.

Donath, J., Karahalios, K., and Viegas, F. (1999) Visualizing Conversations. Proceedings of HICSS 32, Jan 1999.

Ducheneaut, N. (2003). "The reproduction of Open Source software programming communities." Unpublished Ph.D. thesis, U.C. Berkeley.

Eklundh, K.S, and Macdonald, C. (1994) The use of quoting to preserve context in electronic mail dialogues. IEEE Transactions on Professional communication, vol.37, n°4, December 1994.

Eklundh, K. S., and Rodriguez, H. (2004) Coherence and interactivity in text-based group discussions around web documents. Proceedings of the 37th Hawaii international conference on Systems Sciences, 2004

Elliott, M., and Scacchi,W. (2004) Mobilization of Software Developers: The Free Software Movement. To appear in Information, Technology and People, location http://opensource.mit.edu/online_papers.php

Gacek, C., and Arief, B. (2004) The Many Meanings of Open Source. IEEE Software, 21(1), 34-40, January/February 2004.

Herbsleb, J. D., Klein, H., Olson, G. M., Brunner, H., Olson, J. S., and Harding, J. (1995) Object-oriented analysis and design in software project teams. In F. Détienne, and R. Rist (eds): Special issue on empirical studies of object-oriented design. Human-Computer Interaction, 10, 2 and 3, pp 249-292.

Herring, S. (1999) Interactional Coherence in CMC. Proceedings of the 32nd Hawaii Conference on system sciences.

Mahendran, D. (2002) Serpents and Primitives: An ethnographic excursion into an Open Source community. Master's Thesis, School of Information Management and Systems, UC Berkeley, May 2002.

Marcu, Daniel (1997) The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts. Ph.D. Dissertation, Department of Computer Science, University of Toronto, Toronto, Canada, December 1997.

Mockus, A., Fielding, R.T., and Herbsleb, J. (2000) A Case Study of Open Source Software Development: The Apache Server. In proceedings, International Conference on Software Engineering, pages 263-272, Limerick Ireland, June 5-7.

Moran, T. P., and Carroll, J. M. (1996) Design rationale: concepts, techniques and uses, Erlbaum, Mahwah, NJ.

Olson, G.M., Olson, J.S., (2000) Distance matters. Human-Computer Interaction, 15, 139-178.

Olson, G.M., Olson, J.S., Carter, M. R. and Storrosten, M. (1992) Small Group Design Meetings: An Analysis of Collaboration. Human-Computer Interaction, 7, 347-374.

Popolov, D., Callaghan, M., and Luker, P. (2000) Conversation space: visualizing multi-threaded conversation. AVI 2000, Palermo, Italy.

Raymond, E. S. (1999) The cathedral and the bazaar. Available at http://www.tuxedo.org/esr/writings/cathedral-bazaar/

Sack, W. (2000) Conversation Map: A content-based Usenet newsgroup browser. In Proc IUI 2000, ACM Press, 233-240.

Sack, W, Détienne F, Burkhardt, J.M., Barcellini F, Ducheneaut, N, Mahendran D. (2004) A Methodological Framework for Socio-Cognitive Analyses of Collaborative Design of Open Source Software. Distributed Collective Practices workshop in CSCW'04 conference. November 6-10, Chicago, US.

Sandusky, R.J, Gasser, L., and Ripoche G. (2004) Information practices as an object of DCP research, Distributed Collective Practices workshop, CSCW'04 conference, November 6-10, Chicago, US.

Stallman, R. M. (2002), Free Software, Free Society: Selected Essays of Richard M. Stallman, GNU Press.

Smith, M., Cadiz, J. J., and Burkhalter, B. (2000) Conversation Trees and Threaded Chat. Proc. of CSCW 2000, p. 97–105.

Smith, M., and Fiore, A.T. (2001) Visualization Components for Persistent Conversations. Proc. of CHI 2001, 136–143.

Venolia, G., and Neustaedter, C. (2003) Understanding sequence and reply relationships within email conversations : a mixed-model visualization. CHI 2003, April 5-10, Florida, USA.

Viégas, Fernanda B., Marc Smith. (2004) "Newsgroup Crowds and AuthorLines: Visualizing the Activity of Individuals in Conversational Cyberspaces," Proceedings of the 37th Hawaii Conference on system sciences.

Walz, D. B., Elam, J. J., Krasner, H., and Curtis, B. (1987) A methodology for studying software design teams: an investigation of conflict behaviours in the requirements definition phase. In G. M. Olson, S. Sheppard, and E. Soloway (Eds): Empirical Studies of Programmers: Second workshop. Norwood, NJ: Ablex Publishing Corporation. 83-99

Yee, K-P. (2002) Zest: discussion mapping for mailing lists. CSCW 2002 (demo).