# Thematic Coherence and Quotation Practices in OSS Design-Oriented Online Discussions

Flore Barcellini, Françoise Détienne, Jean-Marie Burkhardt, Warren Sack

# Thematic Coherence and Quotation Practices in OSS Design-Oriented Online Discussions

Flore Barcellini, Françoise Détienne
INRIA –CNAM Eiffel team
Domaine de Voluceau BP105
78153 Le Chesnay Cedex France
003313963{5255,5522}
{Flore.Barcellini, Francoise.Detienne}@inria.fr

Jean-Marie Burkhardt
Université Paris 5 LEI
45 rue des Saint-Pères
75270 Paris France
0033142862135
Jean-Marie.Burkhardt@univ-paris5.fr

Warren Sack
University of California Santa Cruz
Film and Digital Dept Multimedia
1156 High Street
Santa Cruz, CA 95064, USA
0011831459-3204
wsack@ucsc.edu

## ABSTRACT
This paper presents an analysis of online discussions in Open Source Software (OSS) design. The objective of our work is to understand and model the dynamics of OSS design that take place in mailing list exchanges. 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 analysis: a thread-based of the reply-to links between messages. The advantages of a quotation-based analysis over a thread-based analysis are outlined. 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.

## Categories and Subject Descriptors
H.4.3 [**Communications Applications**] Electronic mail, H.5.3 [**Group and Organization Interfaces**] Asynchronous interaction, Theory and models.

## General Terms
Design, Human Factors

## Keywords
Distributed asynchronous design, quoting practices, Open Source Software projects

## 1. INTRODUCTION
Open-source software (OSS) design is a particular case of asynchronous, distributed, collaborative design. As analysed previously by Sack et al. [1], the OSS design activity occurs in three activity spaces: the discussion space, the documentation space and the implementation space. 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 large quantity of data generated and archived, proposing methods and tools to extract relevant data, from the design discussions addresses a real need.

In this paper, 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 approach is based on quotation practices which can be used to reconstruct the thematic coherence and to locate design relevant data in online discussion archives. Until now the dominant model used to represent conversation, the threading model, has been based on the reply-to links between messages. Our working hypothesis is that quotation-based representations are more relevant than threading-based representations to reconstruct thematic coherence of design-oriented 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.

In the following sections we review prior work in thematic coherence analysis and in the analysis of quotation in online discussions. We discuss about models and visualisation tools to represent online discussions mainly based on threading. Then we develop our working hypothesis and research strategy. The last two sections of the paper concern the presentation of our study on online discussions in the Python project and the discussion and perspectives.

## 2. 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. A central aspect of thematic coherence in this case is how a message connects to previous messages. 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. Processes of turn-taking and topic (theme)


*GROUP'05*, November 6–9, 2005, Sanibel Island, Florida, USA.

maintenance are subject to disruption and breakdowns. So, some form of explicit (or inferable) link between messages is usually required to understand the thematic coherence of an online discussion.

Herring [2] shows that interactive exchanges in a variety of Computer-Mediated-Communication modes tend to be less tightly stitched together than in face-to-face conversations: responses are often separated from the turns they are responding to, topics tend to decay quickly, and multiple overlapping exchanges often share the same channel. According to a time-based sequential model of on-line conversation (messages are posted in the order received by the system), there are indeed disrupted turn adjacency, i.e. turns that are intended as responses or follow-ups to previous turns, do not occur temporally adjacent to initiating turns [2]: this is a violation of sequential coherence (pragmatic principles of adjacency and relevance). This can create potential confusion that users seek to minimize by adopting compensatory strategies such as conversational linking strategies. Linking is the practice of referring explicitly to the content of a previous message (or previous messages as we will see in our analysis) in one's response.

Eklundh &Macdonald [4] showed that quoting a message, i.e., including it in a comment or reply, was a widely used technique in e-mails dialogues. Quoting is seen as a context-preserving mechanism but the majority of responders use it selectively. Their results showed that users perceived the use of quoting as contributing to the sense of conversation when communicating in e-mail.

On the basis of content analysis, Eklundh & Rodriguez [3] distinguish between several types of conversational linking strategies in on-line conversations around documents:

- Explicit references: message number (in fact, never used), author (e.g. *even through Fred may be right*), subject either by quoting or paraphrasing. Here quoting is seen as a linguistic strategy used by participants to connect a comment to previous discourse contributions.
- Implicit references: deictic or anaphoric reference to previous messages (e.g. *as you mention*), conversational sequencing (question or response move), topic relatedness.
- External references: to other documents, to group experience.

Consequently, quoting is a subtype of linking as an explicit reference.

According to Herring [2], quoting creates the illusion of adjacency: it incorporates portions of two turns within a single message. It maintains context (i.e., portions of previous messages) and so can be used to retrace the history of a conversation.

These studies are based on general topics email or forum discussions. In this paper we are examining the quoting practices in the discussion space of an OSS project: compared to previous studies, these online discussions are oriented by a common group objective which is design.

# 3. THE THREADING MODEL OF ONLINE DISCUSSIONS

Considering the large number of data produce in mailing list and forums, there is a real need of treatment and visualization of on-line discussions. Despite findings on the quoting practices in computer-mediated-communication research, the dominant model in work on online discussions visualization remains the threading model either mixed or not with the sequential (time-based) model. Work in this area ([5], [6]) generally 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.

The threading approach is the main basis of tools for organizing online discussions. Mixed models of visualization combine this approach with the sequential model. Threaded Chat [7] lets users manually attach individual chat messages as replies to others, so that a conversation becomes an evolving tree rather than a scrolling list of messages. With Loom [8] each message is a dot placed on a 2D grid; time is represented on an horizontal axis, authors are represented by rows. Lines connect the dots of a message to its replies. Netscan [9] provides several visualizations for newsgroup discussion structure, most notably time-based thread tree and piano roll views.

These representations are useful to analyse the interactional roles of proponants and repliers in conversations. They are relevant to get a picture of the centrality (versus periphery) of participants in the community of posters: central participants may be considered as those who tend to get more replies to one post (see [17]). However their relevance for identifying and visualising the thematic coherence of online discussion may be questioned on the basis of computer-mediated communication studies enhancing the quoting practices as a major linking strategy.

# 4. OUR WORKING HYPOTHESIS AND RESEARCH STRATEGY

Our working hypothesis is that quotation-based representations are more relevant than threading-based representations to reconstruct thematic coherence of design-oriented online discussions. Whereas the threading model is useful for analyzing interactional roles and for mapping the centrality of participants in a social network (see, for instance [10]), we assume that it is poorer than the quoting model in reconstructing the thematic coherence of online discussions. Quoting, as a linking strategy actively used by posters, connects a comment to previous discourse contributions thus maintaining the thematic coherence in asynchronous discussions. It should thus be a good basis to reconstruct the thematic coherence of the discussion in a more precise way than threading.

We propose to use quoting as the link to extract thematic coherence in online conversations. 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 [11] and a prototype inspired by CONVERSATION MAP called ZEST [12]. 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.

Our research strategy is based on two complementary approaches: "by hand" analysis and "automated" analysis of corpus of design-oriented on line discusions. The by hand analysis is conducted to test the validity of the quoting model to reconstruct the thematic coherence of design-oriented discussions. We compare the quoting graph-model with the threading tree-model by examining how precisely messages belonging to the same design-theme (a design problem) are grouped together in coherent subsets with respect to these two representations.

Based on these results, in an interactive and iterative way, we 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 as in [11]. In this paper we present the "by hand" analysis and discuss the validity of the quoting model to represent online discussions.

# 5. STUDY OF ONLINE DISCUSSIONS IN THE PYTHON OSS PROJECT

## 5.1 Corpus

### *5.1.1 Python project and PEP process*

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.

As Mahendran [13] pointed out in an ethnographic study, the Python project has a very centralized social structure. This characteristic is shared by most of OSS communities: they usually have a strict, hierarchical organization that stratifies developers into levels ([14], [13]). This centralized power structure can be discussed in relation with the freedom ideological-based structure that tend to ground OSS communities ( see [23]).

Mockus, Fielding and Herbsleb [19] assume that when the core team of OSS project is bigger than 15 persons, the project must use explicit means of coordination such as procedures .for setting software evolution. The designers of Python engage in this kind of process, 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 supposed 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 ([15], [16]) have been practiced in many corporate and governmental settings.

In Sack et al [1] 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. Once a rough-draft PEP is accepted by the peps editors (1 administrator and 1 developper, cf. 5.2.2), 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 and his chosen consultants. 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.

### *5.1.2 PEP's discussion*

Our message corpus was drawn from one of the major elements of the discussion space of the Python community: the python-dev mailing list that hosts discussions pertinent to design decisions. The entire conversations are archived on the web and are public.

Up to now, there have been 161 PEPs discussed in the Python project. 28 PEPs are informational PEPs (called meta-PEPs), e.g., a meta-PEP describes the PEP process and 133 PEPs are about new features in Python: 51 PEPs have been accepted and already implemented; 2 PEPs have been accepted but not implemented yet; 26 PEPs have been deferred, rejected, abandonned or withdrawn; 2 peps are empty (abstract); 52 PEPs are open PEPs (under consideration).

We selected conversations regarding two specific PEPs, PEP 279 and PEP 285, which have been accepted and implemented and which were discussed in the same period of time (from March 28th to April 27th of 2002). Thus the Python community structure was approximately the same for these two PEPs.

The PEP 279 corpus 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). The PEP 285 corpus is composed of 2 discussions: part one (96 messages posted by 22 authors between March 29th and April 5th 2002) and part two (23 messages posted by 10 authors between April 3rd and April 9th 2002).

These two PEPs are different according to champion (author of the PEP) and design problem criteria.

Champion: whereas the champion of PEP 279 is a developer, the champion of PEP 285 is an administrator, BDFL.

Design problem type: the two PEPs concern distinctive aspects of the Python language, a function problem (PEP 279) and a type problem (PEP 285).

- PEP 279 proposes three different enhancements to Python: (1) a new index builtin function; (2) a way to facilitate generator comprehension; and, (3) a means for generator exception passing;
- PEP 285 proposes the introduction of a new built-in type, bool, with two constants, False and True.

## 5.2 Method

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 analysis. Each message will be characterized according to these three aspects.

### *5.2.1 Quotation practices and message structure*

Linking strategies may use either explicit references as quoting or implicit references. In the two discussions, there were very few implicit references (3/127 in PEP 279). According to this, coupled with the fact that we want to propose an automatic way to represent discussion, we chose to not take them into account in the following analysis. We have focused our analysis on explicit references and examined how far quoting is a general strategy employed by participants in the PEP discussions.

Each message was categorized according to its structure and the source message(s) that is (are) quoted by the message. The structure was categorized according to the alternation of blocks of quoted material and blocks of commentary (new text) in a message:

- A text-only message, is a message that does not contain any quotations;
- A one-quote message is a message with one block of quotations followed by a comment.
- A multiple-quotes message is a message containing alternating quotes and comments (Mq).

### *5.2.2 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/). One can identify three important, declared roles, related to the PEP process in the Python community:

- The project leader sometimes referred to (semi)-ironically as the BDFL (Benevolent Dictator For Life);
- The core team or administrators: nine people (at the time of our analysis), including BDFL, 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).

### *5.2.3 Message content analysis*

Our message content analysis consisted in identifying the themes addressed by messages.

For PEP 279 discussion, we found five themes corresponding to the following technical design problems:

- T1: this theme concerns the issue of how functions, to be built into the Python language, are to be named; twenty-three alternative names have been proposed;
- T2: different possible syntaxes for the functions have been discussed; eight such syntactic alternatives have been articulated by the discussants;
- T3: it concerns the syntax, semantics and history of a technical issue concerning generator comprehension;
- T4: it concerns the technical issue of generator exception passing;
- T5: it concerns an orthogonal problem of name binding and the status of name spaces (i.e., two other technical issues).

For PEP 285 discussion, we found six main themes corresponding to the following technical design problems:

- T1: this theme deals with the consequences of a new built-in type, bool, on the Python language;
- T2: it deals with a specific function ,str, the status of the variable (boolean or integer) that is returned and its implication for backward compatibility;
- T3: it concerns the name of the constant of the new built-in type. The issue is whether it should be named like in Java or in C99;
- T4: it concerns the elimination of non-boolean operations on bools;
- T5: it concerns a specific operator of Python and what it should return, a Boolean or an integer;
- T6: it concerns the inheritance relationship between Int and Bool.

## 5.3 Results

### *5.3.1 Quotation practices and message structure*

Table 1 shows the distribution of message structures in PEPs 279 and 285 discussions. We can note that the distribution of type of messages does not differ between the two discussions (Khi2=2.278, DoF=2, p=.32). Quoting is a general strategy employed by participants since 92% (PEP 279) to 96% (PEP 285) of messages are either one-quote messages or multiple-quotes messages.

**Table 1. : Frequencies of types of messages in PEP 279 and PEP 285 discussions**

| Messages | PEP 279 | PEP285 |
|---|---|---|
| Text-only | 8% (11/127) | 4% (5/119) |
| 1-quote | 69% (87/127) | 71% (85/119) |
| M-quote | 23% (29/127) | 25%(29/119) |

### *5.3.2 Quotation practices and thematic coherence*

Our analysis of quotation practices allows us to compare a representation of online discussion in PEPs 279 and 285 based on quotation-based links between messages (Figure 1b and 2b) with a representation based on threading or "reply-to" links between messages (Figure 1a and 2a). In the figures, the circles or squares 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 or squares are displayed differently to represent the theme (i.e., the different design problems enumerated above) addressed by the messages.

Figure 1a and Figure 2a display an analysis of the discussion based on the 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 several 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 1b and Figure 2b display an analysis of the discussion based on quotation links between messages. It reveals a distinctly different organization of the messages. For example, in Figure 1b, four areas can be discerned: at the beginning of the conversation, the four themes (T1, T2, T3 and T4) are treated simultaneously in the messages (black circle) except for two messages that discuss only T2. Immediately thereafter two themes, T1 and T4 become the foci of discussion. Finally, an orthogonal problem, T5, emerges.

The thematic coherence of the discussion, especially regarding T1, is better represented by the quotation-based links of Figure 1b and Figure 2b than by the reply-to links of Figure 1a and Figure 3a. In this quotation-based analysis all of the messages are connected together, compared to the distinct threads shown in Figure 1a and Figure 2a (3 distinct threads for PEP 279, 6 threads for PEP 285). In this analysis almost every message is linked to another message and the thematic coherence of the discussion is preserved. There are only a few text-only messages (3 for PEP 279, 2 for PEP 285) that needed to be linked to the others using a reply-to relationship.

Closer examination of the message contents reveals that the messages that are unlinked in Figure 1a and Figure 2a are pivotal to the overall discussion. Here is an example, of message 4 for PEP 279 that summarizes design alternatives and their rationales and call for new rationales; and generates several branches of discussion. (quotation are preceeded by ">".)

*"After some more thinking about the name, I have two contenders left: enumerate() and indexer(). Let me explain why I reject the others:*

*>iterindexed()—five syllables is a mouthfull.*
*Indeed.*
*>index() -- nice verb but could be confused the .index() method*
*Indeed. [...]*
*So now I'd like to choose between enumerate() and indexer(). Any closing arguments?"*

By comparing the position of messages 4 and 68 in Figure 1a with their positions in Figure 1b, one can see that the reply-to representation does a poor job of positioning them where they should be. Figure 1a shows messages 4 and 68 in detached and peripheral positions. In contrast, Figure 1b, 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.

### *5.3.3 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 discussions. The geographically-distributed nature of the project makes this an important issue to study. The results are displayed in Table 2.

**Table 2. : Temporal distribution of 1st quotes in PEP 279 and PEP 285 discussions**

| | PEP 279 | PEP285 |
|---|---|---|
| 50% of 1st quote appears within | 1 hr | 2hr16 |
| 75% of 1st quote appears within | 5hrs | 7hr33 |

According to Table 2, 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. Indeed, we observed that half (median) of the 1st quotations of posted messages come within 1h (PEP 279) or 2h16 (PEP 285). Furthermore, 3/4 of the quotations occur within 5h (PEP 279) to 7h33 (PEP 285). It means that responses quoting a message are quickly posted regarding the distributed nature of the community. Late citations are often posted by co-designers who are far away from the US (where most participants are) and their messages then arrive after design decisions have been taken.

### *5.3.4 Discussion participation and assigned roles in the Python project*

The distribution of participants within developers and administrators categories is similar in the two discussions (Khi2=.246, DoF=4, p=.993). Table 3 displays their contribution to the discussion in terms of percentage of messages posted.

**Table 3 : Percentage of messages posted for each category of participant in PEPs 279 and 285 discussions**

| Status | PEP 279(part 1) | PEP285(Part 1) |
|---|---|---|
| BDFL | 17% | 19% |
| HP-A | 23% | 8% |
| LP-A | 3% | 3% |
| HP-D | 40% | 57% |
| LP-D | 17% | 13% |

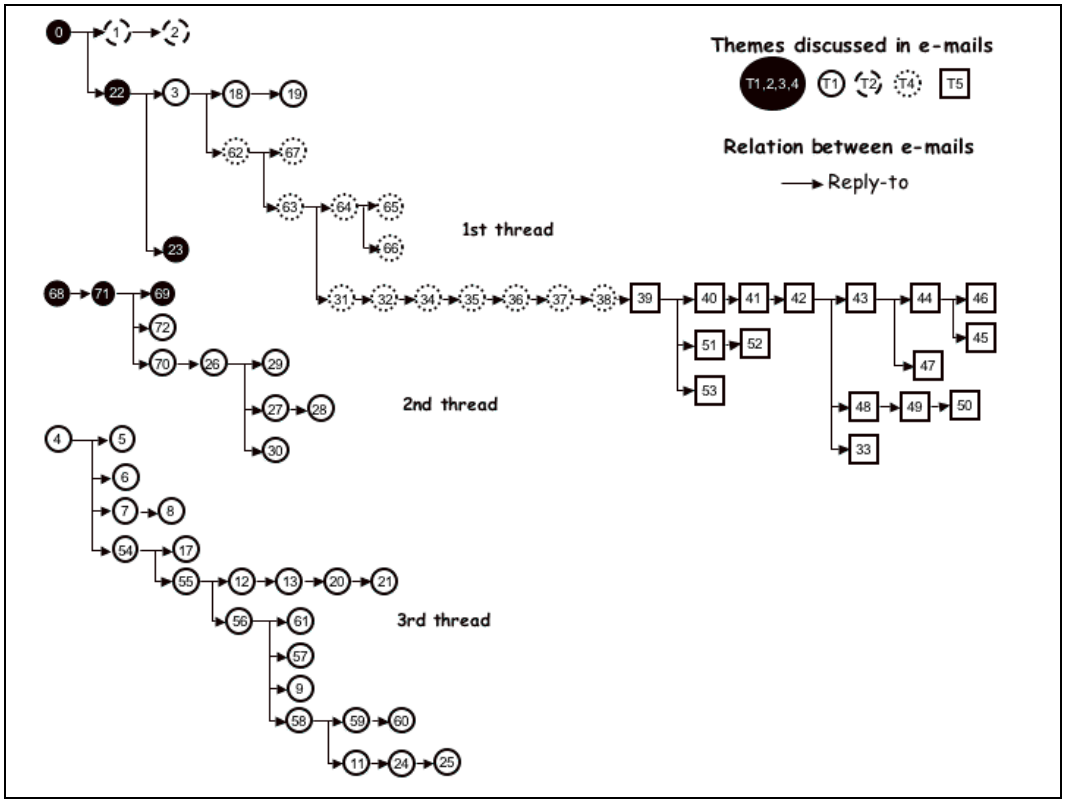


**Figure 1a: Threading based representation of the links between messages PEP 279 (part 1)**

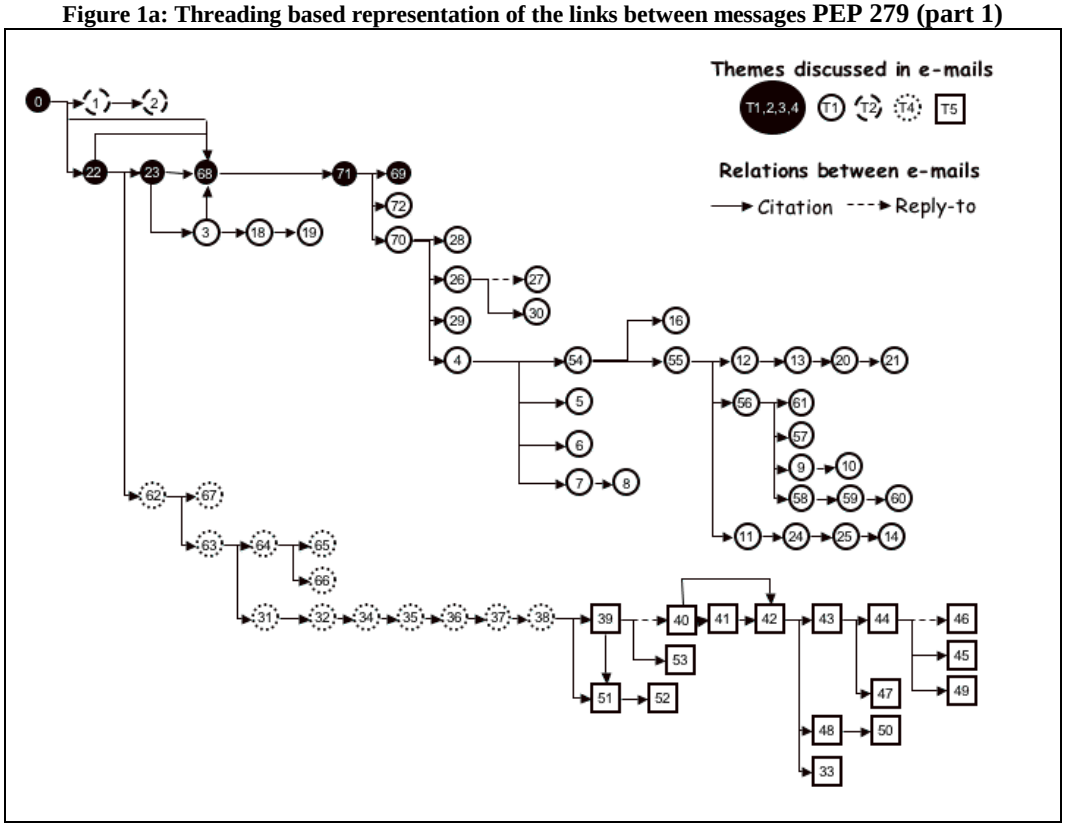


**Figure 1b: Quotation-based representation of the links between messages PEP 279 (part 1)**

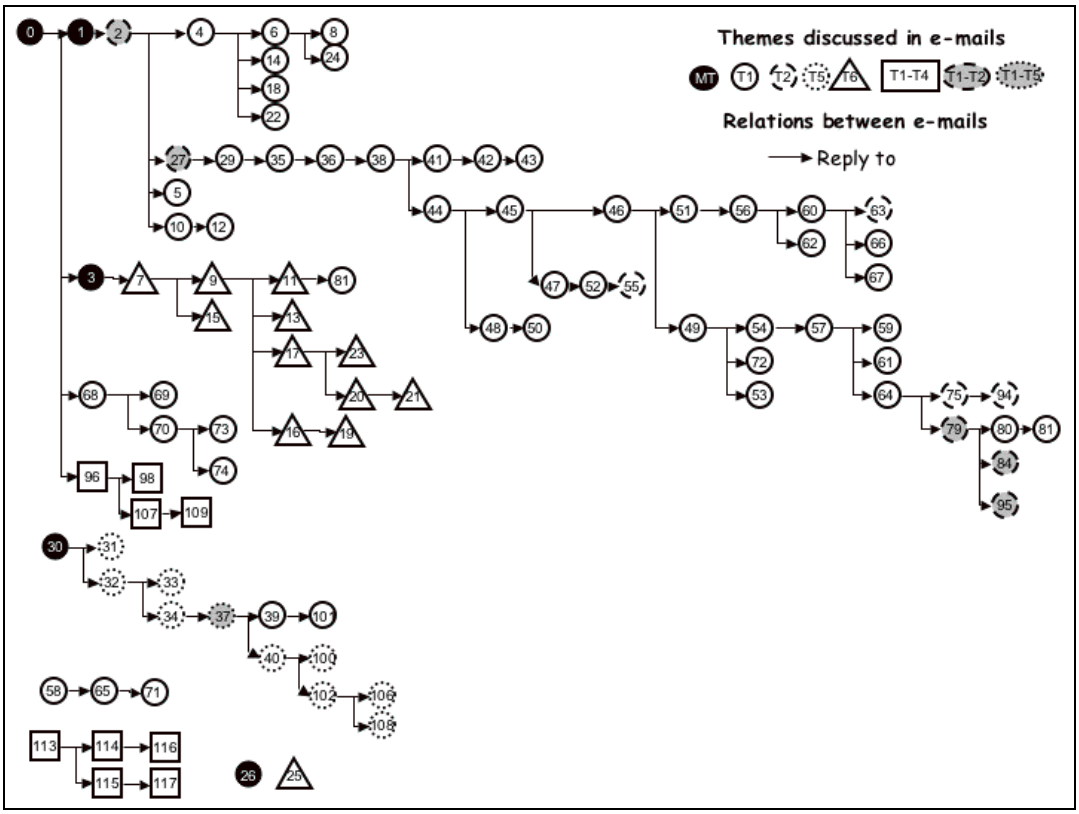


**Figure 2a: Threading based representation of the links between messages PEP 285 (part 1)**

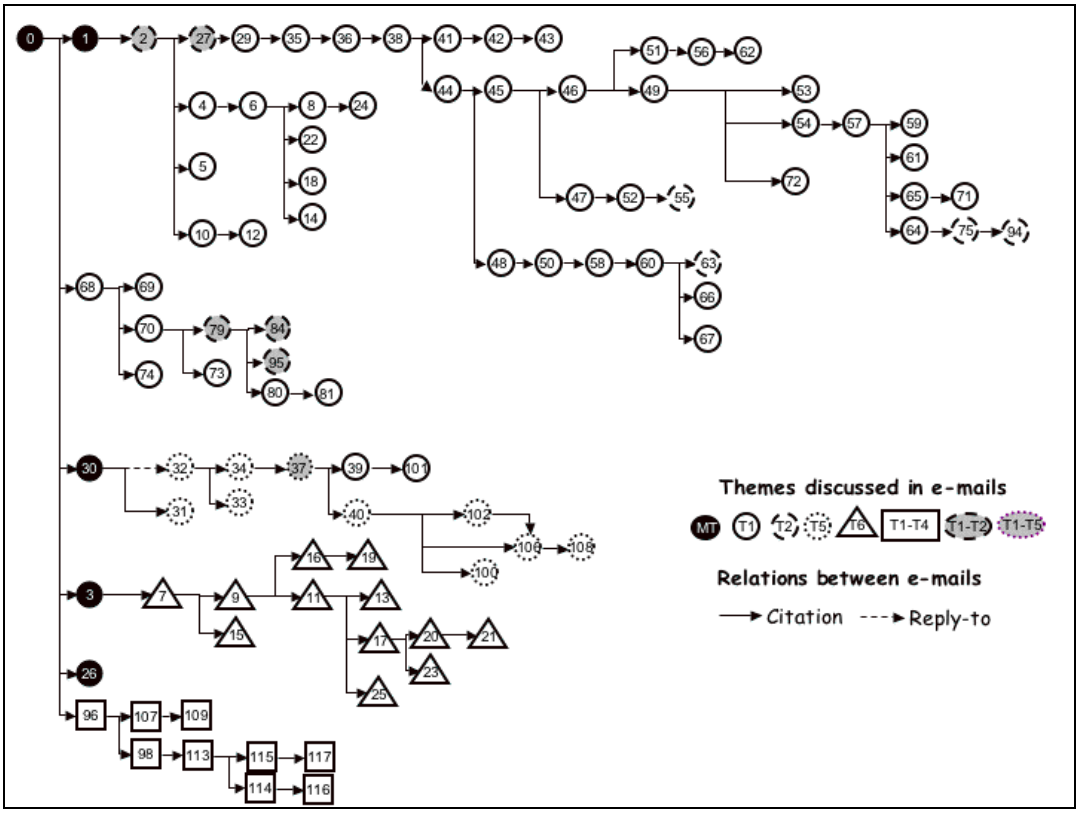


**Figure 2b: Quotation-based representation of the links between messages PEP 285 (part 1)**

However, the distribution in terms of percentage of posted messages differs from PEP 279 to PEP 285 (Khi2=10,882, DoF=4, p<.03). Indeed, we can notice that administrators (especially HP-A) are much less present (as they posted less messages) in the PEP 285 discussion compared to the PEP 279 discussion; conversely, developers (especially HP-D) are much more present. This may be explained by the fact that BDFL is the champion of PEP 285 and the other members of the core group (administrators) trust him in leading the discussion and the decision process and in taking the final decisions. Also it seems that some discussions within the core group for PEP 285 take place outside the dev-list: some messages mention private email discussions. This is not the case for PEP 279. A complementary explanation may be the type of design problems which is addressed in PEP 285: the introduction of a new built-in type in Python is tightly coupled with other Python design tasks and this may encourage more participations of developers.

We completed this analysis visualizing the position of participants in the discussions. Figure 3 and Figure 4 display discussions graphs where messages are labelled with the project roles of their posters. Figure 3 (PEP 279) 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 T5 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 4 shows another structure of discussion for PEP 285. BDFL is still strongy present, he posted 19% of the messages and he is present at all positions in the graph (at the beginning of a branch, closing message and sequential position). Developers are more present (70% of posted messages) and are also present at every position in the graph. HP-A are present only in the discussion around theme 1 which is a theme around the implication of the new built-in function for the Python language.

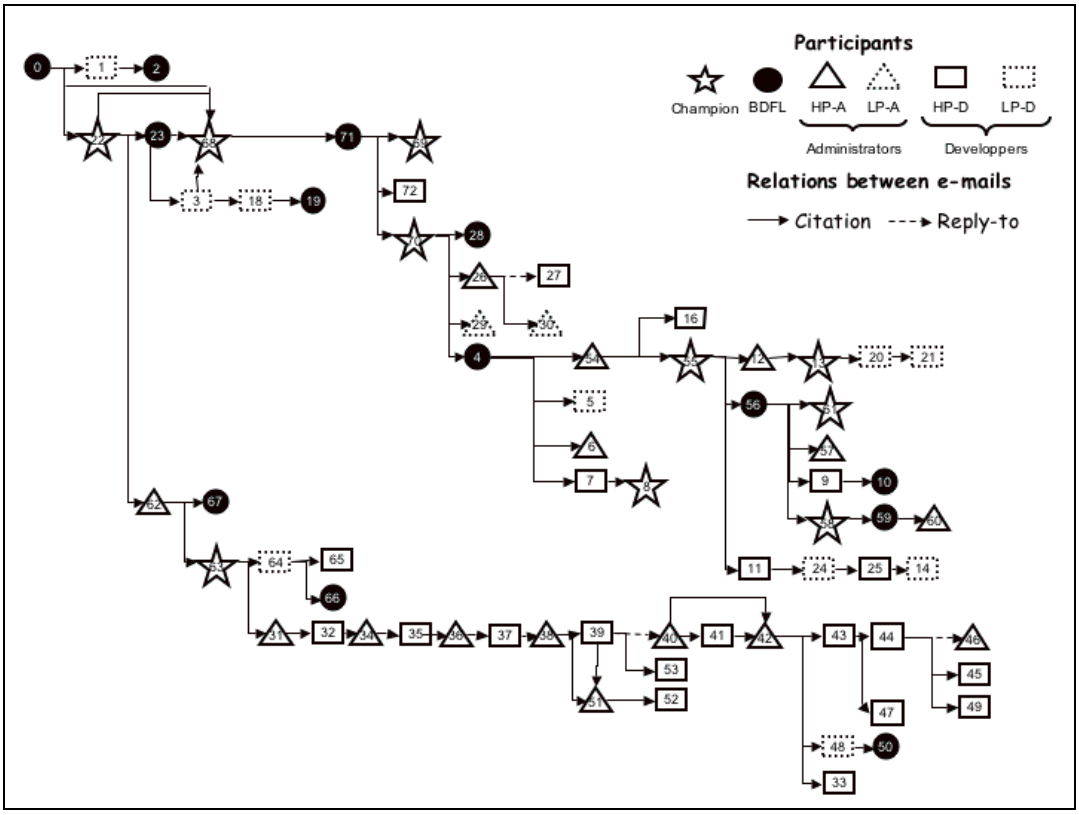


**Figure 3: Status and position in the discussion PEP 279**

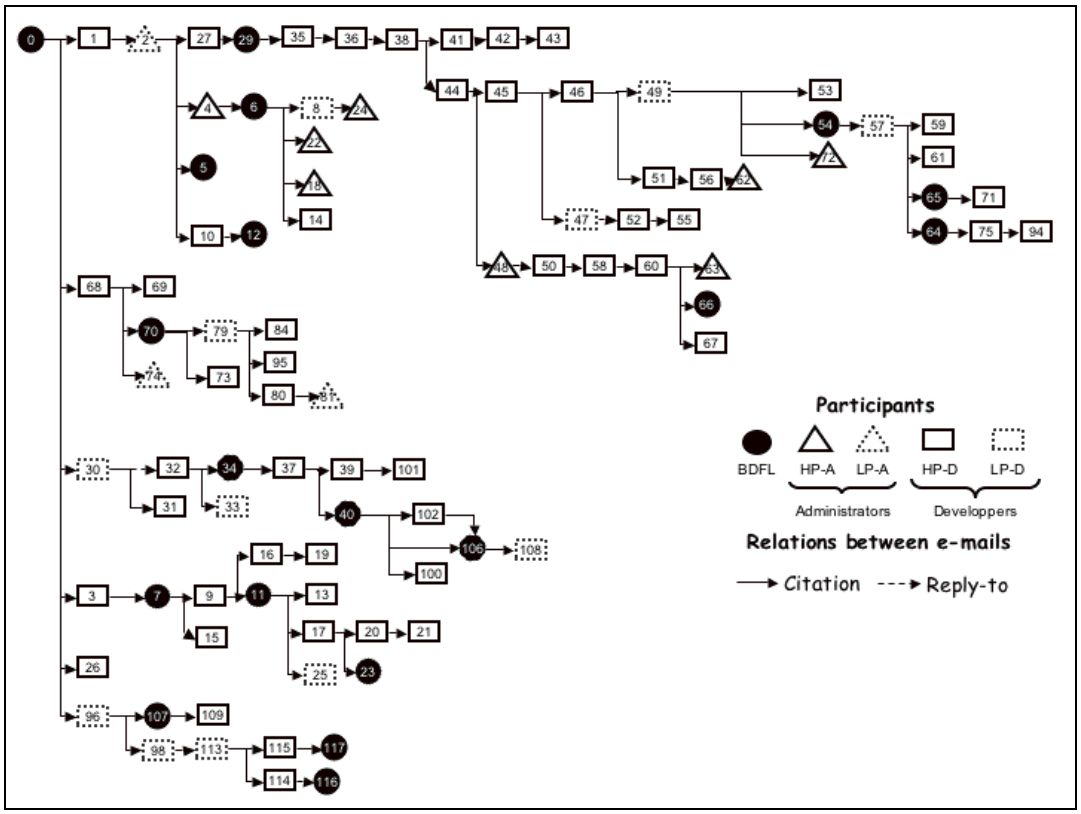


**Figure 4: Status and position in the discussion PEP 285**

## 6. DISCUSSION

Previous studies of OSS design projects have focused on different activity spaces. Mahendran's ethnographic work [13] illustrates how power is distributed across the three activity spaces - the discussion, implementation and documentation spaces. Ducheneaut's work [17] 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 (which is one of the major Open-Source project). Sandusky et al. [18] focused their analysis on the documentation space of the Bugzilla project. Mockus et al. [19] focused their analysis on the implementation space of this same project. We focus on the discussion space.

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 discussions analyzed. A quotation-based analysis did not exhibit these weaknesses.

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. This was particularly clear in one of the PEP discussions we analysed where two participants led the discussion: 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.

## 7. PERSPECTIVES

Our study is an analysis of two PEP discussions. 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, [20]). Further work will be done on other discussions, systematically varying these factors .In parallel we keep developing a tool to automate some parts of the analysis.

Our long-term perspective is to retrace the design-rationale of the OSS design, as it had been proposed and done in traditional design process (see for example [22]). This will be based on our quotation-based analysis coupled with message content analysis of messages. Indeed, we have started to characterize the message content with respect to categories of design activity reflecting the rhetorical function of the message. This analysis is developed and discussed in another paper (see Barcellini et al. [21])

We intend to build on the quotation analysis procedures currently incorporated in the Conversation Map system [11], thereby, to provide some automated means to foster knowledge sharing in distributed collective practices. We also hope to construct software to automatically analyse themes of discussion computing, and, to analyze patterns of argumentation, an admittedly much more difficult task akin to rhetorical structure parsing ([24]).

## 8. 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.

## 9. REFERENCES

[1] 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. Paper presented at the Distributed Collective Practices workshop *in CSCW'04.* November 6-10, Chicago, US.

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

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

[4] 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, (pp197-202).

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

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

[7] Smith, M., Cadiz, J. J., and Burkhalter, B. (2000) Conversation Trees and Threaded Chat. In *Proceedings. of CSCW 2000*, p. 97–105.

[8] Donath, J., Karahalios, K., and Viegas, F. (1999) Visualizing Conversations. In *Proceedings of HICSS 32*, Jan 1999.

[9] Smith, M., and Fiore, A.T. (2001) Visualization Components for Persistent Conversations. In *Proceedings. of CHI 2001*, 136–143.

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

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

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

[13] 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.

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

[15] 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.

[16] 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.

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

[18] Sandusky, R.J, Gasser, L., and Ripoche G. (2004) Information practices as an object of DCP research. Paper presented at the Distributed Collective Practices workshop in *CSCW'04*. November 6-10, Chicago, US

[19] Mockus, A., Fielding, R.T., & Herbsleb, J. (2002). Two Case Studies of Open Source Software Development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology, 11(3),* 309-346.

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

[21] Barcellini, F, Détienne, F., Burkhardt, JM., Sack, W. (2005). A study of online discussions in an Open-Source community : reconstructing thematic coherence and argumentation from quotation practices. In Van Den Besselaar, P., De Michelis, G., Preece, J., Simone, C. (Eds) *Communities and Technologies2005 (pp 301-320),* Dortmund, The Netherlands, Springer.

[22] Moran, T. P., Caroll, J. M. (1996) *Design rationale: concepts, techniques, and use*. Mahwah, NJ, USA: Laurence Erlbaum Publisher.

[23] Elliott, M., and Scacchi,W. (in press). Mobilization of Software Developers: The Free Software Movement. *Information, Technology and People.*

[24] 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