HUMAN TECHNOLOGY

An Interdisciplinary Journal on Humans in ICT Environments ISSN: 1795-6889

www.humantechnology.jyu.fi Volume 7 (1), May 2011, 30–48

INDIVIDUALS' USE OF E-MAIL COMMUNICATION GENRES IN OPEN SOURCE SOFTWARE COMMUNITY BUILDING

Vitaliano Barberio
Department of Management
University of Lugano
Switzerland

Antonio Mastrogiorgio
Department of Management
University of Bologna
Italy

Alessandro Lomi
Department of Management
University of Lugano
Switzerland

Abstract: *The advent of the participative Internet (Web 2.0) sheds a new light on traditional knowledge about communication practices. The role of information and communication technologies seems to be very central in managerial literature, while the human side of the issue is less considered. This paper argues that individuals establish communication genres as semantic templates for accomplishing their communicative projects. Communication genres are codes of default behavioral expectations resulting from recurrent communication actions over the time. By using semantic network analytical techniques, our argument is explored in a particular empirical setting, that is, a virtual community of open source software development.*

Keywords: *communication genres, semantic networks, open source, cognitive models, vote practices.*

INTRODUCTION

The recent evolution of the Internet (i.e., Web 2.0) has brought about an unprecedented unfolding of distributed collaborative communities that specify new business models, ICT tools, and products. Open source software (OSS) projects are emblematic examples of these new human–technology interaction forms, where individuals cooperate at a distance in order to jointly complete tasks and achieve common objectives. In 1999 Eric Raymond introduced the metaphors of the "bazaar" and the "cathedral" that highlight the dramatic difference between OSS development and commercial software development.[1] In order to contribute to the general knowledge about individuals' use of communication in OSS projects, this paper builds on three related ideas common in managerial and sociological literatures. First, organizational production

URN:NBN:fi:jyu-201152310898

has general coordination principles, and OSS communities do not differ. Second, repeated communicative actions let individuals institutionalize communicative forms, or *genres*, providing a framework for the contents they share. Third, once established, communication genres can be reinforced by sanctioning nonconforming behaviors.

The main objective of this paper is to explore in depth the practical implications for individuals using Internet-related communication genres. A case study is presented about software developers voting by e-mail. The case has been extracted from mail repositories of an OSS project, with communication practices consolidated over 12 years of community development history. More precisely, we selected the Apache project, whose development community counts hundreds of contributors worldwide. Because of the geographical dispersion, Apache's developers coordinate with each other via several Internet-based tools—e-mail, chat, code repositories, wikis, issue trackers—rather than face-to-face. The case articulates a semantic network analysis of e-mail contents belonging to a community-building (not technical) mailing list. Two particular e-mail threads, whose messages' sequential arrangement shows a content-genre conformity issue, were analyzed. Semantic network analysis allowed us to structurally evaluate the issue of a genre's conformity in terms of its impact on shared mental models.

The next section offers a short review on coordination in OSS projects, highlighting a lack of attention to the human side of the human–technology dyad. We follow, then, with the theoretical argument for the ICT-mediated process, going from individual cognition to communication genre and back to individuals. Next, we describe methods and data used in our case study, and follow that with the results of the case. The paper ends with our conclusions and proposals for possible extensions of this research.

LITERATURE ON COMMUNICATION IN OSS COMMUNITIES

Literature on OSS projects has flourished over the last 10 years. Focusing on the particular topic of individuals' use of communication, we have been able to distinguish two main lines of reasoning. The first one is referred to here as *contingent*. It portrays organizational communication primarily as a means of information exchange to enable coordination between interdependent product development tasks (Baldwin & Clark, 2006; von Hippel & von Krogh, 2003). The second perspective on organizational communication in OSS is considered *institutional*, and portrays communication as the manifestation of underlying status and trust social dynamics (Chen & O'Mahoney, 2007; O'Mahoney & Ferraro, 2007).

The contingent perspective is prevalent in management and software engineering literatures. From a theoretical point of view, it entails a model of organizations as information processing systems (March & Simon, 1958), where individuals use communication and ICTs in order to coordinate interdependent tasks in conditions of uncertainty (Daft & Lengel, 1986; Thompson, 1967). In OSS communities where neither formal authorities nor central planners are responsible for task assignment, many communicative behaviors are expected to mirror product architecture (Sanchez & Mahoney, 1996). Following this reasoning, and since software has a more modular architecture than other products, virtual communities producing software will tend to represent more distributed communication structures than other productive organizations (Baldwin & Clark, 2006; von Hippel & von Krogh, 2003). Considering the nature of economic incentives to participate, von Hippel (2007) proposed that open source

communities can be thought of as flat networks of user nodes linked by information exchange. According to this view, the content of communication is the mutual assistance that programmers provide each other. Kuk (2006) proposed that programmers use communication as means of epistemic search for the knowledge that they need in order to solve their technical problems. In doing so, they try to interact with other programmers who control more valuable knowledge, but also accept a general rule of reciprocity.

The institutional perspective, which is more visible in sociological studies on organizations, entails communication reflecting status and trust dynamics arising from cultural processes (Powell & DiMaggio, 1991). Community members, by means of repeated interaction, are expected to institutionalize shared codes and values to create what sociologists call an organizational field (DiMaggio & Powell, 1983). Thus communities are ensembles of actors who share a collectively constructed reality (Berger & Luckmann, 1966). Empirical research in this strand displayed encouraging results from our point of view. For example, Chen and O'Mahoney's (2007) ethnographic study of four virtual communities showed that communication among members operates a synthesis between two logics over the time: (a) an expression logic, which advocates informal organizations as a way to respect differences in members' motivations, abilities, timeliness, and accountability and to encourage broad participation; and (b) a production logic, which endorses rationalized, bureaucratic practices, such as a division of labor and rules. In a study on the Linux Debian project, O'Mahoney and Ferraro (2007) used content analysis techniques on electoral candidacy e-mails in order to explore the concept of meritocracy as shared within the development community. They showed that such a shared understanding shifted over the time from a more technical (code-writing) concept of leadership to a more organizational one (community-building). Then, inverting the logic, they compared individual performance and the likelihood of being appointed to a community-management position and found the likelihood of a leadership position was affected by the individual's behavioral conformity to the socially rewarded kind of merit (technical or organizational). Communication as repeated interaction among community members seems to provide a means of social evaluation and trust building, resulting over time in status-consolidated social structures. For example Grewal, Lilien, and Mallapragada (2006) have shown that individuals' embeddedness (Uzzi, 1996), measured as centrality in communication networks, increases legitimacy, and thus positively affects access to resources and performance.

The main strength of contingent approaches (Baldwin & Clark, 2006) is their recognition of interdependence-related aspects of individuals' participation in organizational work. However, this line of research has not paid much attention to the human side of the human–technology dyad on the one hand, and to communication contents, on the other hand. The institutional research has insightfully pointed out that socially institutionalized codes enhance coordination (or reduce uncertainty) by constraining the scope of possible actions (O'Mahoney & Ferraro, 2007) and the access to resources (Grewal et al., 2006). However it seems to us that a microanalytical exploration of the human–technology interaction process is lacking in both the theoretical and empirical literature. Moreover from a methodological point of view, semantic analysis techniques could usefully integrate the content analysis methods we have seen employed so far for revealing shared meaning structures. Our research shares the basic assumption of institutional logic trying to deal with the three highlighted gaps. Therefore, in order to understand how individuals deal with communication practices in OSS communities, the next three sections will (a) model existent (theoretical) contributions into a conceptual

microlevel interaction framework; (b) propose a semantic network analysis method for mining shared meaning structures (methodological); and (c) provide empirical substance to both the theoretical and methodological claims (empirical).

THEORETICAL FRAMEWORK

In Figure 1 the general framework of our research is illustrated. It is modeled as a human–technology interaction cycle with several aspects. In (Step 1), individuals make sense of reality by arranging symbolic elements into internal representations. Next (1→2), individuals' participation in groups entails the use of communication for establishing shared understandings and achieve coordination. Over time (Step 2), individuals' participation in groups makes communication genres emerge as templates for enhancing contents' contextualization. Finally, (2→1), genres are reinforced and modified by means of use over time. An organization provides technological and symbolic context (Area 3) for the interaction. The concepts in this paragraph concern the ideal link between an individual's perception of work and ICT-mediated communication genres. For the sake of simplicity, we refer to the OSS development process as an organizational setting. However, many ideas behind the model actually possess validity in general human–technology interaction.

From consolidated research in cognitive psychology (Rosch, 1975), we know that human beings act differently according to their own internal representations of the world. Each individual holds several such representations, known as cognitive models (CMs), which are selectively activated according to the situation. Even if some deep cognitive structures are very stable within any given individual, CMs are expected to evolve over the time. Because CMs are semantically reflected within the social use of language (Lakoff, 1987), this idea of internal representations also applies insightfully to the representation of coordination within

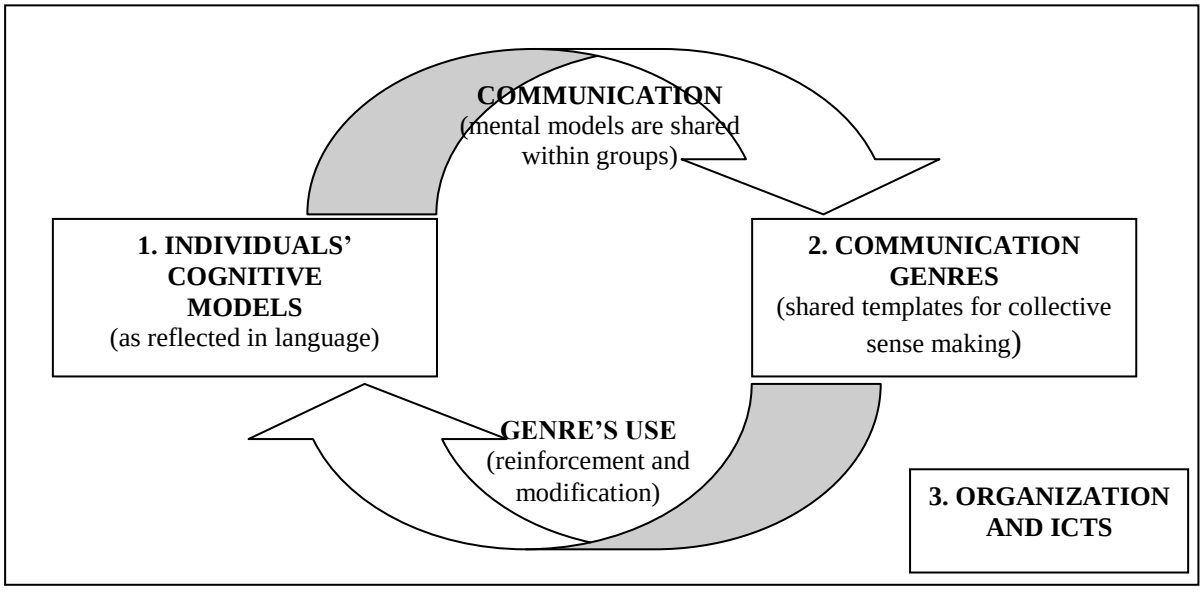

Figure 1. General framework: The relationship between individuals, groups, and communication genres in organizational human–technology processes.

on-line communities. In fact, due to the scarcity of or even the frequent absence of face-to-face interaction, virtual community members rely primarily on e-mail (and other communication tools) for expressing their opinions and connecting with their peers.

OSS development is characterized by communities where members take part in a socio-organizational productive activity. Technical tasks concerning software development are managed mainly by means of code repositories and issue/bug trackers (Mockus, Fielding, & Herbsleb, 2002). Hence, we can expect that, when engaging in e-mail communication, the developers deal with the distinct need for achieving a shared view on their actions. In other words, it is still a matter of coordination; however, rather than only technical tasks, now there are different and sometimes contrasting individuals' representations of the situation (either technical or social) that need to be composed. Through communicative actions, individuals share their CMs, enabling collective sense making and problem solving. Because collective decision-making forms of organizing encourage diversity, they permit a wider range of organizing practices than a conventional form would allow (Weick, 1995). Since community members may develop competing ideas about the best way to organize (Chen & O'Mahoney, 2007), the communication content is expected to reflect the selective process of such practices.

Organizational forms of production provide a both material–technological and symbolic context to communication leading to the emergence of institutionalized practices of communication referred to in this paper as *communication genres* (Yates & Orlikowski, 2002). Drawing on Giddens' (1984) structuralist perspective, Orlikowski and colleagues (Im, Orlikowski, & Yates, 2005; Yates & Orlikowski, 1992) proposed an approach for studying organizational communication that is based on genres (e-mails, meetings, expense forms, reports, etc.) as social structures constituted through individuals' ongoing communicative practices. Communication represents an ideal domain in which the sociocultural dynamics comes into being, arising from the practices of different communication genres (Im et al., 2005; Yates & Orlikowski, 2002). As these authors suggested in a recent work regarding exploring communication genres in commercial software development,

> These genres are socially recognized types of communicative actions that are habitually enacted by organizational members over time to realize particular social purposes in recurrent situations (Yates and Orlikowski, 1992). Through such enactment, genres become institutionalized templates that shape members' communicative actions. Such ongoing genre use, in turn, reinforces those genres as distinctive and useful organizing structures for the community. (Im et al., 2005, p. 5)

What, then, is the content of the feedback relation connecting communication genres and individuals' CMs? Here we propose that communication genres can be thought of as metacognitive models, which are shared schemas for reassembling individual cognitive models into collective understandings. Generally speaking, structuration theory concerns the production, reproduction, and transformation of social institutions that are enacted by the use of social rules (Giddens, 1984). These rules shape the action taken by individuals in organizations. At the same time, by regularly drawing on the rules, individuals reaffirm or modify the social institutions in an ongoing, recursive interaction. Whether used explicitly or implicitly as organizing structures, genres shape beliefs and actions, both enabling and constraining how an organization's members engage in communication (Im et al., 2005).

METHOD

Setting

Because we wanted to go in depth with the exploration of genre structures' enactment in e-mail communication, OSS development seemed an ideal empirical setting. In many OSS projects, people taking part in product development never or very rarely meet face-to-face. We can then consider discussion threads unfolding in mailing lists a symbolic construction of the reality available to participants to link the (software) artifacts with the community.

We have chosen to focus on the Apache OSS project for our empirical investigation. The Apache project started in February 1995 when Rob McCool stopped developing his httpd-server software at the National Center for Supercomputing Applications in the USA, and a small group of users, the so-called Apache Group, began a combined effort to coordinate fixes to the existing code. After several months of adding features and small fixes, the Apache Group replaced the old server code base in July 1995 with a new architecture designed by Robert Thau. The core developers were distributed around the world and all of them were working on the project as volunteers. Therefore, both the leadership and coordination mechanisms were distributed as well, to take into account the limited time that each programmer could devote to the project. As one of the founding members pointed out,

> Unlike most open source projects, Apache has not been organized around a single person or primary contributor.... There was no Apache CEO, president, or manager to turn to for making decisions. Instead, we needed to determine group consensus, without using synchronous communication, and in a way that would interfere as little as possible with the project progress. What we devised was a system of voting via email that was based on minimal quorum consensus. Each independent developer could vote on any issue facing the project by sending mail to the mailing list with a "+1" (yes) or "-1" (no) vote." (Fielding, 1999, p. 42)

The Netcraft Survey (April 2011)[2] currently lists Apache as the most important server software in the world by market share (179,720,332 websites served across all Internet domains, 61% of total), followed by Microsoft (57,644,692 websites served, 25% of total). Actors in Apache's development constitute a little universe clustered around more than 100 subcommunities. Like other consolidated OSS projects that spent their early years with a straight focus on technical issues, the Apache community only slowly began addressing social organizational issues. Apache now has very consolidated practices and even several community building guidelines[3] that indicate,

> Mailing lists are the life blood of Apache communities. They are the primary mode of discourse and constitute a public and historic record of the project. Other forms of communication (P2P, F2F, personal emails and so on) are secondary… The reason is that communications on other than the public mail aliases exclude parts of the community. Even publicly advertised IRC chats can be exclusionary due to time zone constraints or conflicting time commitments by community members who might want to participate. (The Apache Software Foundation, 2009, Communication, para. 1)

The so-called Apache Way is an interesting cultural code for studying individuals' involvement in new Internet-based collective efforts. Given a very simplified picture,

decisions are taken in two ways: (a) consensus generation around a given proposal by means of simple discussion, and (b) voting on the emergent proposal when no consensus is reached by means of simple conversation. To keep the distributed decisional process moving, each Apache voting system lasts no more than 72 hours.

Data Collection and Subjects

Data were gathered from an infrastructural mailing list belonging to the Apache Software Foundation (instituted in 1999). The community mailing list[4] was created in 2002, after a period of institutional reorganization (1999–2002), with the aim of discussing topics regarding community-building. At the time of our study the target mailing list had more than 300 subscribers around the world. We selected a particular discussion concerning the "if" and eventually the "how" the community mailing list itself should be made accessible to the participants.

The discussion comprised 155 individual e-mails clustered within two threads. The temporal lag of the discussion on both threads extended from 22 October, 2002 to 6 November, 2002. All of the retrieved e-mails had the tag [vote] in its subject field. Following previous research on e-mail genres (Im et al., 2005), this would entail all of the mail belonging to the same communication genre. Because we wanted to investigate some potential concern with the use of genres, these two threads were selected as involving a voting session on the same issue (same subject plus same tag = [vote]), but extending over time for considerably longer than the usual 72 hours (the Apache Way).

While most scientific papers address specifically the demographic characteristics of its study participants, this would be a nearly impossible task in our study. OSS Projects such as those used for this study often do not collect basic demographic characteristics on the participants. Moreover, if they are collected, they may not be complete. Finally, not all members of the list participate in all of the discussions, so the sample is not representative in any event. Specifically, we did not have access to personal data on the list members, and thus cannot provide that information here. We recognize that this situation presents a limitation to the generalizability of our findings.

Network Text Analysis

Semantic network analysis, also known as network text analysis (NTA), is used in this paper to explore the template structure provided to the emerging shared understanding within the communication genres. NTA is a set of analytical methods for applying network techniques to the semantic analysis of shared meanings extracted from texts. Like content analysis techniques, NTA can be used to extract mental models from text (Carley & Palmquist, 1992). However the main difference between content analysis and NTA is that while the former considers only the frequencies of isolated concepts, the latter also takes into account the semantic connections between concepts. A semantic network is a graph where the nodes are concepts and the links are inferred as relationships between the concepts. For all automatic processing steps on e-mail texts and later on for semantic network analysis, the CASOS[5] software suite has been used.

The e-mail texts have been processed in order to extract semantic network reconstructions of individuals' CMs. Natural language texts extracted from the e-mails went

through a set of procedures entailing coding choices about what concepts to delete or generalize (Carley, 1993). This preprocessing work entailed three main steps. The first step deleted redundant information within the original set of 155 e-mails. In fact, when answering messages (Reply), people often included the quoted text from the original message. Text blocks, or *corpora*, quoted across messages could have some relevance on emerging semantics, but that was not essential for revealing elementary meaning structures. Hence, for the sake of simplicity in this study, we deleted them. The second step refers to noise deletion. Every single word is characterized by a frequency of occurrence within each single text (individual frequency) and across all texts (aggregated frequency). The distribution of words' frequencies seems to follow a power law distribution, that is, a few words present high frequencies of occurrence while the most of words occur just a few times. However the concepts that occur most often are usually irrelevant to the extent of reconstructing meaning structures. These (often trivial) concepts are normally related to grammar and syntax, such as articles, pronouns, and prepositions, and so these "irrelevant" words were deleted. Very low-frequency words—those occurring once or twice across all texts—represent idiosyncratic concepts also not expected to be relevant in the domain of discourse, and so they were deleted as well. This preprocessing step applied a thesaurus file on the selected (postnoise deletion) set of concepts. In doing so, we made the grain of the text coarser by bringing back similar concepts to a single, more general concept: The loss of information represents a gain in terms of synthesis. The third and final preprocessing step was the inference of relationships between concepts. The software tool we used for this purpose includes an automatic words–proximity procedure that addresses the issue of link generation based on the co-occurrence of words within a "window" that slides over the text (Carley, 1997; Danowski, 1982).[6] According to this procedure, the more two concepts occur in a proximal position within texts the more intense the semantic relation is between them.[7]

The outcome of the preprocessing steps is a set of 155 semantic networks, one for each e-mail corpus. These semantic networks are CMs' representations made by nodes and links, where the nodes are concepts and links are the relationships between such concepts inferred from their proximity within text corpora. The weight of a link between two nodes has been set as their co-occurrence frequency. All semantic network analysis will, of course, be developed by comparing results from two separate clusters of e-mails, that is, one cluster for each mailing list thread. Two semantic analysis techniques have been used for further analyzing the extracted cognitive models: semantic connectivity and visual map analysis. Semantic connectivity (Carley & Kaufer, 1993) allows the detection of important concepts recurring across multiple texts. It entails a linguistic classification of concepts, just like nodes in semantic networks, based on their centrality (Wasserman & Faust, 1994) and consensus across distinct texts. In this research we focused only on *symbols*, which are concepts with a combined high score[8] comprising both degree and betweenness centrality, on the one hand, and on consensus (the number of texts in which it occurred) on the other. Finally a visual map analysis was conducted in order to make as clear as possible the role of genres on shared meanings. In practice we created an aggregate semantic network containing the union set of concepts for all 155 individual semantic networks and union set of links (sum of weights criterion). The resulting graph was simplified by removing links with weight less than a threshold of 5 (co-occurrence frequency) to improve visual interpretation.

RESULTS

Semantic Networks Descriptive Statistics

Semantic network descriptive statistics computed across all 155 semantic networks are reported in Table 1. We notice that e-mail texts in Thread 1 presented a higher average number of concepts with a higher standard deviation than e-mails in Thread 2. Semantic networks in Thread 2 are slightly more densely connected (0.06) than networks in Thread 1 (0.05). Moreover semantic networks in Thread 1 displayed a higher average clustering coefficient than those from Thread 2. This means that e-mails within the first set tended to form clusters or, in other words, to have reciprocally connected neighborhoods than those in second set do. These facts tell us that the e-mails in Thread 1 are, on average, characterized by larger and less cohesive sets of concepts than the e-mails in Thread 2.

Symbols

The semantic connectivity analysis procedure gave as output a classification of concepts from which we selected the 10 highest scoring symbols (see Table 2) for further interpretation in this paper. By confronting the 10 highest ranking symbols for each thread, we were able to draw some additional comparative results. First, in Thread 1, the concept View scores fourth in rank, immediately after Apache, Committers (the core developers), and Vote. We interpret this as a manifestation of the discussion focus around a sort of call to committers to express their personal views and, at the same time, a call for voting. Second, in Thread 2, the concept View is not present among the top 10 symbols. Instead, the concepts Vote 1 and Vote 2 are important because they express a polarization of voting around two precise proposals. Third, it is noticeable that the negative vote designation (expressed in Table 2 as -1) also ranked very highly. As we discovered by looking closer at the communication contents, this happened because most of the committers voted against (-1) the proposal to fully open (reading and writing) the community mailing list. Both threads have a committer's name in the top 10 symbols, "Sam" in Thread 1 and "Roy" in Thread 2. This could be interpreted as the detection of two distinct "voices" (or points of view) whose presence in discourse was respectively more evident in each of the threads. In reviewing the original messages, we were able to confirm that

Table1. Statistics Across all Semantic Networks from Both Thread 1 and Thread 2 (N = 155).

	Thread 1 (n = 88)				**Thread 2** (n = 67)			
Measure	**Min.**	***M***	**Max.**	***SD***	**Min.**	***M***	**Max.**	***SD***
Number of concepts	5	37.31	113	21.89	5	20.45	54	7.87
N. of isolated concepts	0	0	0	0	0	0.97	4	0.87
Number of links	4	46.33	162	32.14	2	21.87	76	10.42
Density	0.01	0.05	0.20	0.03	0.03	0.06	0.17	0.02
Clustering Coefficient	0	0.03	0.08	0.02	0	0.02	0.17	0.03

Table 2. Concepts Classified as Symbols (Scoring High or Over the Mean in Degree, Betweenness, and Consensus).

Thread 1 (74 concepts in this class)					**Thread 2 (**50 concepts in this class**)**				
Rank	**Concept**	**Consensus**	**Degree Centrality**	**Betweenness Centrality**	**Rank**	**Concept**	**Consensus**	**Degree Centrality**	**Betweenness Centrality**
1	Apache	0.01	0.09	0.11	1	Vote	0.01	0.06	0.24
2	committers	0.02	0.09	0.10	2	vote1	0.04	0.09	0.13
3	Vote	0.01	0.08	0.11	3	Need	0.00	0.02	0.21
4	View	0.01	0.07	0.09	4	Committers	0.03	0.06	0.12
5	Archive	0.01	0.08	0.08	5	vote2	0.04	0.08	0.07
6	Org	0.01	0.06	0.09	6	-1	0.03	0.02	0.13
7	community	0.01	0.06	0.06	7	let's	0.04	0.02	0.12
8	Sam	0.01	0.05	0.06	8	Community	0.00	0.04	0.13
9	More	0.01	0.05	0.05	9	Roy	0.00	0.03	0.13
10	Do	0.01	0.06	0.04	10	No	0.03	0.02	0.10

*Notes***.** *Consensus* refers to a measure of the extent to which concepts recur across distinct pieces of text. The *degree centrality* of a given node (concept) is the number of its direct connections with other adjacent ones, expressed as a fraction of all possible connections between dyads of nodes in a network. The *betweenness centrality* of a node measures the extent to which it rests in all (shortest) paths connecting all possible dyads between other nodes in a network (Freeman, 1979). All measures are rounded to two decimal places.

"Sam" was speaking more in favor of an open policy of access, while "Roy" was the spokesperson of the party proposing a more restrictive policy.

Consolidated Semantic Networks

In order to refine preliminary findings of the semantic connectivity analysis illustrated above, we have constructed a consolidated semantic network (CSN) for each mail thread. A CSN relative to a thread is intended as the union of all the semantic networks corresponding to the e-mails forming that thread. The sum criterion was used to unite the networks. This means that a link between two nodes in the CSN has weight equal to the sum of link weights across all individual networks presenting that dyad. In order to enhance the visual interpretability of results, all links having a weight less than 5 (co-occurring less than 5 times in a thread as a whole) were removed from the data set. Isolated (totally disconnected, degree = 0) concepts where removed iteratively as well.

In looking at the CSN generated for Thread 1 (Figure 2), it is clear that the concept View is the most central in terms of weighted degree centrality (the most of high-weight links are related to it). Starting from the node View, we can find at least two paths that could be interpreted as roughly corresponding to the alternative proposals involved in the voting process:

(a) *view → close → except → committers → members → invitees*

(b) *view → open → completely → anyone → can → subscribe → post → read*.

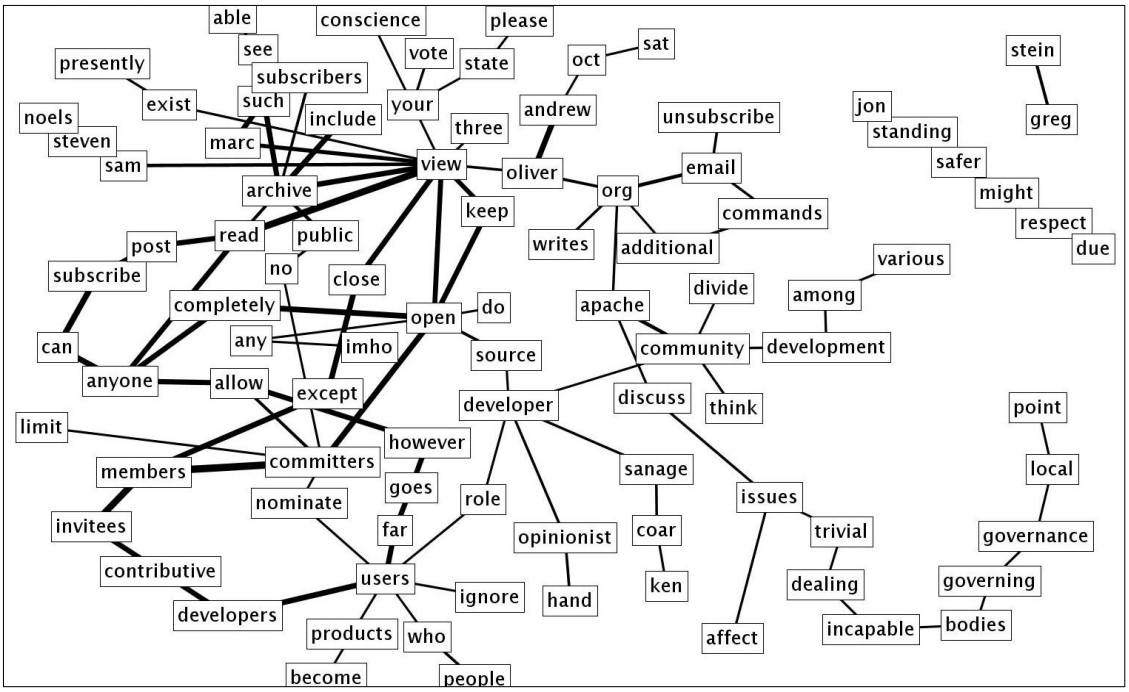

Figure 2. A consolidated semantic network (CSN) from e-mail Thread 1. Links with a weight less than 5 and isolated nodes have been removed from the original network to offer a clearer representation. Links' width reflects weights, that is, the frequency of co-occurrence for a dyad of concepts in the thread.

The first proposal entailed opening the mailing list only to active participants and those invited by current committers, while the second proposal entailed fully opening the list to anyone interested. Another interesting path is located in the bottom right side of Figure 2:

> (c) *local → governance/governing → bodies → incapable → dealing → trivial → issues → affect.*

This semantic path or sequence of concepts seems to express that some developer perceived the potential scenario where everybody is allowed to read and write on the mailing list as a threat. The marginal position of this "tail" to the main component does not necessarily mean that the underlined meanings are of secondary importance. In fact it could be interpreted as the fact that, even if few participants used the concepts in this cluster, the underlined meaning attaches directly to core arguments in discussion. From a discourse point of view, we could also think that the position expressed by this path is very clearly understandable for an audience, while very central nodes could be characterized by more ambiguity and noise.

As we have shown so far (see Tables 1 and 2), the CSN based on Thread 2 data is characterized by a less complex[9] semantic structure. This simpler structure makes more recognizable the basic elements of the voting dynamic (the +1 and -1 concepts) entailed by e-mails. In major detail we can see (in Figure 3) the symmetric structure of options in Thread 2, making it clearer. For example, let us consider the most central the e-mail-mediated voting process. Concepts like Vote1 and Vote2 refer to kinds of formalized proposals to be voted: nodes 1, 0 and -1 are the actual manifestations of voting preferences. Departing from those

concepts/nodes, we can find again, even if in a more stylized representation, the elements corresponding to two different proposals:

(a) *vote1 → 1 → yes → let's → open → everyone*

(b) *vote2 → committers → keep → private*.

The content of communication concerned "who" should be allowed to do "what." The options for the who issue were expressed by the concepts Committers and Non-committers, while the options available for the issue were Write and Read. Different configurations of these elements were formalized initially within a proposal with alternative scenarios to be voted on by community members in Thread 1. Because a second voting session (Thread 2) took place around the same subject, we expected that some kind of realignment of the communicative action would have been proposed. So we double-checked for concrete proof of this *genre tuning* within the content of e-mails. Not surprisingly, we found in the very first e-mail within Thread 2 (see Figure 4) a claim for switching to a more structured and linear voting session because the previous consensus formation process was too complex for univocal interpretation.

The e-mail in Figure 4 also provides a concrete idea about the flexibility of the vote genre in adapting to the situation. In response to the individual points of view that emerged in the preceding session, the second voting clarified two points: there is no need to provide reasons for the vote, and to simplify the voting, since only full numbers were sought.

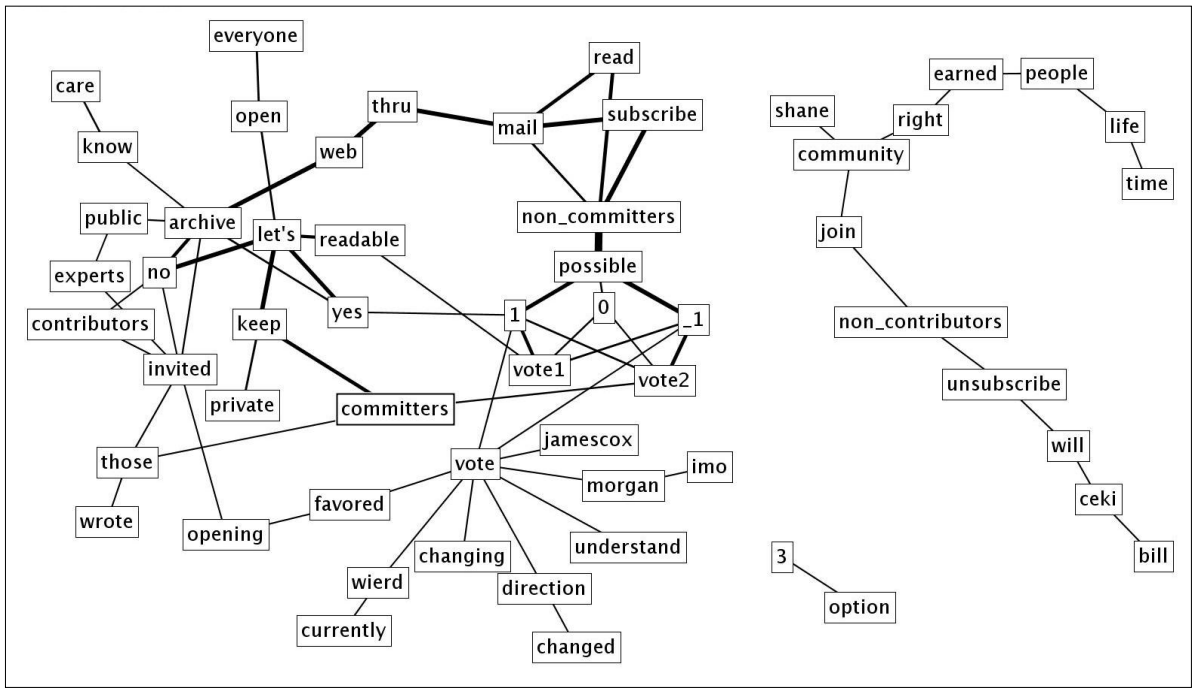

Figure 3. Consolidated semantic network from e-mail Thread 2. Both links with weigh less than 5 and isolate nodes have been recursively removed from the original network to offer a clearer representation. Links' width reflects weights, that is, the frequency of co-occurrence for a dyad of concepts in the thread.

```
Please, allow me to restart the voting in order to make it easier to reach some
consensus since it's hard to interpret the results of the previous one.
                                - o -
There are two different concerns for openness:
  - open to read
  - open to write
Let's try to keep them separate.
NOTE: there is no need to indicate the reasons for your votes, either
for negative ones.
Also, please, don't vote 0.5 or other numbers, let's keep it simple for
the final count.
                                - o -
VOTE 1:  would you like to make it possible for non-committers to read
this mail list through a web archive?
  [ ] +1 yes, let's make it readable
  [ ]  0 don't know/don't care
  [ ] -1 no, let's keep it private
                                - o -
VOTE 2:  would you like to make it possible for non-committers to fully
subscribe to this mail list?
  [ ] +1 yes, let's open it to everyone
  [ ]  0 don't know/don't care
  [ ] -1 no, let's keep it for committers only
                                - o -
Please, place your vote even if you already voted in the previous poll. We'll reset
the clock and give 78 hours for the vote. I volunteer to count the results and post
them here.Thanks in advance and sorry for the double poll noise.
```

Figure 4. The body of text from the first (in chronological order) e-mail in Thread 2. It should be noted that the author of the above message mentions a 78-hours time period instead of the 72-hours that a careful reader of this article could have expected. This was a mistake in the original text that we reproduced as is. A statement resetting the timing to a 72-hours time window followed later in the e-mail discussion.

DISCUSSION AND CONCLUSIONS

The Apache Way as a system of coordination and governance practices based on shared values (an institution) inspired the worldwide OSS community in the past and still continues to provide an example for many new OSS projects. It structures distributed decision making and coordination through enacting several communication genres (e.g., vote, bug, cvs, proposal, etc.). In the Results section, we detailed some aspects relative to the use of the vote genre. In particular, we showed the interaction between the vote communication genre and collective mental models (represented as CSNs) as activating alternative scenarios within the developers' collective mind. One contribution of this paper is the understanding of dialogic forms of coordination as enabled by the use of communication practices. We have seen that collective decision making with e-mail voting systems entails two main steps: consensus creation and voting. Voting is not mandatory but it saves time when consensus cannot be reached by simple discussion. These two communicative actions could be contextual or sequential. Moreover, as we have seen in the case of the double poll, even the most consolidated practices can be varied according to the situation.

These facts make us consider the important differences existing between compliance to formal rules and the use of communication genres as alternative coordination tools. Even if

the setting of our study was specific (OSS development), the results identified could be of interest in the more general research on coordination practices. The core principles of practices are not a set of conscious, constant rules, but rather practical schemes, often opaque to their possessors, and vary according to the logic of situation (Bourdieu, 1990). Faraj and Xiao (2006) suggested that, in complex knowledge and fast changing environments, the "lens of practice" is more suitable to understanding coordination than traditional contingent approaches. When work situations are characterized by novelty, unpredictability, and ever-changing combinations of actors, tasks, and resources, it could be very complicated to specify ex ante systems of routines and formalized plans of action (Faraj & Xiao, 2006).

In such contexts, then, the concept of trajectories (Strauss, 1993), the sequences of actions toward a goal, could better emphasize the interplay between the contingencies and interactions among actors. Trajectories better allow a consideration of eventual deviations within the course of action from the desired objective. In those scenarios, decisional processes deal more with the situation rather than with formal organizational arrangements. Mische and White (1998) proposed the concept of conversation as a discursive form in which the story does not lend itself to a precise final, and the concept of situation as regarding the possibility of an unexpected or problematic final (which, in our case, was the absence of a clear policy for community management). In summary, it seems to us that communication practices provide a perfect lens for looking at coordination in a concrete way. Moreover, the usefulness of such a lens is increased by the progressive affirmation of dialogic–cooperative forms of coordination enabled by new Internet technologies.

A second contribution of this paper is the network representation (i.e., the CSN) of shared understandings. This analytic tool allowed us to show the reduction of cognitive complexity underlined in collective decision making and coming from the re-orientation of communication genres in use. We have seen how this worked in practice in our case study. If the first thread resulted in a more complex and fuzzy outcome because it was more focused on consensus formation, the second thread effectively reset the situation toward a final decision. This result contributes to recent literature on communication genres and communication management. For example, Luckmann (2009) stated that communicative genres may be defined by their function as models for the solution of specifically communicative problems of social life.

ENDNOTES

1. The bazaar metaphor entails a distributed-production system, involving a large number of developers and characterized by (a) the absence of a centralized decision-making unit defining the ex-ante direction of development of the software code; (b) parallel design and debugging; (c) the integration of users into the production of software code; (d) self-selection of programmers (volunteering) for the tasks that best match their abilities. The cathedral metaphor entails an opposite approach. Here software development is compared to a cathedral-building process carefully crafted by individual wizards or small bands of mages working in splendid isolation, with no beta to be released before its time.

2. The most recent Netcraft survey and its archives are publically available at http://news.netcraft.com/archives/2011/04/06/april-2011-web-server-survey.html

3. Find Apache community-building guidelines at http://incubator.apache.org/guides/community.html

4. The Apache community list is available at http://mail-archives.apache.org/mod_mbox/www-community/

5. For parsing texts and creating semantic networks, we used AUTOMAP. For semantic network analysis procedures, we used the software ORA. Both tools are available at http://www.casos.cs.cmu.edu/ for free use, with conditions.

6. For example, a window of size 2 sliding over the text "John writes code" produces the two distinct associations "John → writes" and "writes → code."

7. Of course, the larger the window, the more noise is captured in terms of nonmeaningful inferred relations. On the other hand, if the window is set too small, important semantic relations are not captured. See Carley (1997) for more details on windowing/proximity procedures.

8. A concept is considered as scoring high in a given measure when its value for that measure is one standard deviation over the mean of the corresponding distribution across all the 155 networks in the sample.

9. A more detailed explanation on semantic complexity is given in the Appendix.

REFERENCES

Apache Software Foundation, The. (2009). Guide to successful community building (DRAFT). Retrieved May 2, 2011, from http://incubator.apache.org/guides/community.html

Baldwin, C., & Clark, B. (2006). The architecture of participation: Does code architecture mitigate free riding in the open source development model? *Management Science, 52*, 1116–1127.

Berger, B. L., & Luckmann, T. (1966). *The social construction of reality*. New York: Doubleday.

Bourdieu, P. (1990). *The logic of practice*. Stanford, CA, USA: Stanford University Press.

Carley, K. M. (1993). Coding choices for textual analysis: A comparison of content analysis and map analysis. *Sociological Methodology*, *23*, 75–126.

Carley, K. M. (1997). Extracting team mental models through textual analysis. *Journal of Organizational Behavior, 18*, 533–538.

Carley, K. M., & Kaufer, D. S. (1993). Semantic connectivity: An approach for analyzing symbols in semantic networks. *Communication Theory, 3*, 183–213.

Carley, K. M., & Palmquist, M. (1992). Extracting, representing, and analyzing mental models. *Social Forces, 70*, 601–636.

Chen, K. K., & O'Mahoney, S. (2007). The selective synthesis of competing logics [MIT working paper]. Available at http://web.mit.edu/iandeseminar/The_Selective_Synthesis_of_Competing_Logics_3_3_07.pdf.

Daft, R. L., & Lengel, R. H. (1986). Organizational information requirements, media richness and structural design. *Management Science*, *32*, 554–571.

Danowski, J. (1982). A network-based content analysis methodology for computer-mediated communication: An illustration with a computer bulletin board. *Communication Yearbook, 6*, 904–925.

DiMaggio, P. J., & Powell, W. W. (1983). The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. *American Sociological Review, 48*, 147–160.

Faraj, S., & Xiao, Y. (2006). Coordination in fast-response organizations. *Management Science, 52*, 1155–1169.

Fielding, R. T. (1999). Shared leadership in the Apache project. *Communications of the ACM, 42*, 42–43.

Freeman, L. C. (1979). Centrality in social networks: Conceptual clarification. *Social Networks, 1*, 215–239.

Giddens, A. (1984). *The constitution of society*. Berkeley, CA, USA: University of California Press.

Grewal, R. G., Lilien, L., & Mallapragada, G. (2006). Location, location, location: How network embeddedness affects project success in open source systems. *Management Science*, *2*, 1043–1056.

Im, H. G., Orlikowski, W., & Yates, J. (2005). Temporal coordination through communication: Using genres in a virtual start-up organization. *Information Technology & People*, *18*, 89–119.

Kuk, G. (2006). Strategic interaction and knowledge sharing in the KDE Developer mailing list. *Management Science*, *52*, 1031–1042.

Lakoff, G. (1987). *Women, fire and dangerous things*. Chicago: Chicago University Press.

Luckmann, T. (2009). Observations on the structure and function of communicative genres. *Semiotica, 173*, 267–282.

March, J. G., & Simon, H. A. (1958). *Organizations*. New York: Wiley.

Mische, A., & White, H. (1998). Between conversation and situation: Public switching dynamics across network domains. *Social Research*, *65*, 695–772.

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*, 309–346.

O'Mahoney, S., & Ferraro, F. (2007). The emergence of governance in an open source community. *Academy of Management Journal, 1*, 1079–1106.

Powell, W., & Di Maggio, P. (1991). *The new institutionalism in organizational analysis*. Chicago: University of Chicago Press.

Raymond, E. S. (1999). *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. Sebastopol, CA, USA: O'Reilly & Associates.

Rosch, E. (1975). Cognitive representations of semantic categories. *Journal of Experimental Psychology, 104*, 192–223.

Sanchez, R., & Mahoney, J. T. (1996). Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management Journal*, *17*, 63–76.

Strauss, A. L. (1993). *Continual permutations of action*. New York: Aldyne de Gruyter.

Thompson, J. (1967). *Organizations in action*. New York: McGraw-Hill.

Uzzi, B. (1996). The sources and consequences of embeddedness for the economic performance of organizations: The network effect. *American Sociological Review*, *61*, 674–698.

Von Hippel, E. (2007). Horizontal innovation networks by and for users. *Industrial and Corporate Change*, *16*, 293–315.

von Hippel, E., & von Krogh, G. (2003). The private-collective innovation model in open source software development. *Organization Science*, *14*, 209–223.

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge, UK: Cambridge University Press.

Weick, K. E. (1995). *Sense making in organizations*. Thousand Oaks, CA, USA: Sage.

Yates, J., & Orlikowski, W. (1992). Genres of organizational communication: A structurational approach to studying communication and media. *Academy of Management Review*, *17*, 299–326.

Yates, J., & Orlikowski, W. (2002). Genre systems: Structuring interaction through communicative norms. *Journal of Business Communication*, *39*, 13–35.

Authors' Note

The present research is part of a broader project on "Models for representing organizational knowledge," supported by MIUR (The Italian Ministry of University and Scientific Research) through the FIRB research funding scheme (grant code number RBNE03A9A7005).

All correspondence should be addressed to:
Vitaliano Barberio
Department of Management Science,
University of Bologna
via Capo di Lucca 34,
40100, Bologna, IT
vitaliano.barberio@gmail.com

Human Technology: An Interdisciplinary Journal on Humans in ICT Environments
ISSN 1795-6889
www.humantechnology.jyu.fi

APPENDIX

This appendix shows in major detail how the implementation of a strict communication genre (i.e., voting) could reduce the complexity of meaning structures underlined in discourse. As discussed on p. 34, conversation (as a low-structured dialogic practice) is a good way to express diverse opinions. At the same time, it could be very difficult to make effective and efficient decisions relying just on conversational practices. We expected that shifting from a conversation to a voting session could be a solution for collective decision making because it reduces discussion complexity. In terms of semantic networks, we could formulate that as the simplification of a high number of almost randomly interrelated concepts into a smaller number of more structured ones.

In Table A1 we report some structural indicators of overall complexity for consolidated semantic networks representing both Thread 1 (left column) and Thread 2 (right column). Indicators are computed first on the "original" consolidated networks (a) that consider all of the existing concepts, and then on the "reduced" networks (b) that consider the same networks after removing links with weight less than 2 (concepts that co-occurred less than twice across all e-mails within the same thread). Figures A1 and A2, below, graphically illustrate the effects of such reduction over semantic networks.

Table A1. Structural Indicators of Overall Complexity for Consolidated Networks.

	Thread 1		Thread 2	
Measure	**Original (a)**	**Reduced (b)**	**Original (a)**	**Reduced (b)**
Node count	360	345	306	61
Link count	1639	887	631	72
Density	0.013	0.008	0.007	0.020
Characteristic path length	3.813	5.086	6.296	5.296
Clustering coefficient	0.076	0.047	0.048	0.034
Network levels (diameter)	14	17	17	12
Degree centralization	0.076	0.050	0.089	0.057
Betweenness centralization	0.102	0.161	0.237	0.117
Closeness centralization	0.237	0.019	0.038	0.025

In comparing the original networks (Figure A1, where *t* represents the respective thread), we see that Thread 1 (a) is more complex than Thread 2 (b) in terms of concepts/nodes, 360 (t1) versus 306 (t2), and number of ties, 1,639 (t1) versus 631 (t2). Moreover, when observing the reduced networks (Figure A2), this difference is accentuated dramatically. In fact, when the data undergoes a reduction process, Thread 1 (a) has 345 nodes and 887 ties, while Thread 2 (b) has 61 nodes and 72 ties. So when we did not consider concepts with a very low co-occurrence frequency across all of the e-mails (which are more likely to be "noise"), we see that the number of nodes just slightly decreases in Thread 1 (-4.2%), but dramatically fall in Thread 2 (-80.1%). In a similar direction, the links in Thread 1 decrease almost by half of the total (-45%), whereas the links in Thread 2 decrease by almost as twice that (-88.6%).

(a) t1, original **(b) t2, original**

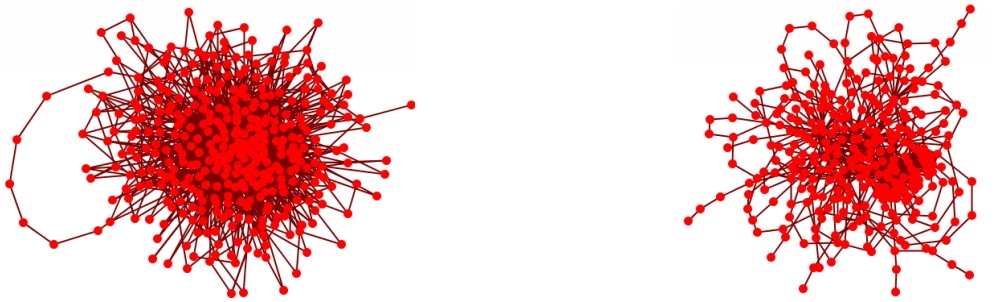

Figure A1. Original consolidated semantic networks (before link/node reduction). The network on the left (a) refers to the content of Thread 1. The network on the right (b) refers to the content of Thread 2. Isolated nodes are not displayed for the sake of clarity.

(a) t1, reduction **(b) t2, reduction**

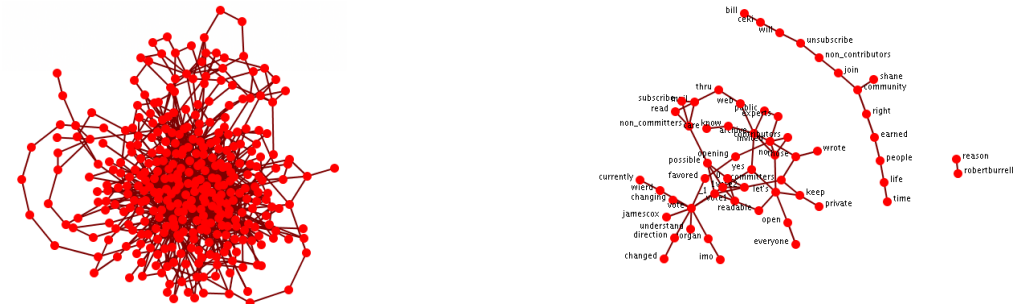

Figure A2. Reduced consolidated semantic networks (after link/node reduction). The network on the left (a) refers to the content of Thread 1. The network on the right (b) refers to the content of Thread 2. Isolated nodes are recursively hidden for the sake of clarity.