A Comparison Study on the Coordination Between Developers and Users in FOSS Communities

Shinsuke Matsumoto, Yasutaka Kamei, Masao Ohira and Ken-ichi Matsumoto
Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara, Japan
{shinsuke-m, yasuta-k, masao, matumoto}@is.naist.jp

ABSTRACT

Recent studies have been trying to better understand geographically distributed software development which is increasing from year to year. Analyzing Free/Open Source Software (FOSS) communities would help us obtain useful insights for distributed software development, because some FOSS communities already have tremendous success in a distributed environment. Our prior study analyzed an informal social structure in the Apache community. In this paper, we describe a comparison study which analyzes informal social structures in the Apache community and the Netscape community, with a focus on participants who belong to both developer and user groups and who assist the collaboration between them. We perform a static analysis for looking at an informal social structure in a single period and a dynamic analysis for observing the transition of the structure over time. As a result, we have found that one of the key factors for the success of distributed software development such as FOSS development was the existence of highly motivated participants who help developers and users collaborate closely with each other.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Collaborative computing, Computer-supported cooperative work, Web-based interaction*

General Terms

HUMAN FACTORS

Keywords

FOSS community, geographically distributed software development, informal social structure, static and dynamic social network analysis

1. INTRODUCTION

Software development in Free/Open Source Software (FOSS) communities has received more attention in recent years, since successful FOSS communities have been developing high reliability and high performance software. FOSS development is an example of geographically distributed software development in which developers around the world communicate with each other. Analyzing a communication structure in FOSS development communities would help us better understand distributed software development and gain useful knowledge of it.

Many studies reported analyses of actual communications among developers in FOSS communities [1, 2, 9, 10, 15]. Howison et al. analyzed an informal social structure formed by developers in FOSS communities [2, 9]. They found that many FOSS projects had a few core developers who participated in a project for substantial periods of time and that the central participants in large projects did not change. Bird et al. examined the correlation between centralities and Apache developers' contributions from developer mailing lists and the change history of source codes [1]. The analysis results indicated that developers communicating with many other developers contributed further to source code changes.

While many studies observed developers' communication, [19, 24] suggested the importance of users' roles and collaboration between developers and users in FOSS development. Raymond et al. [19] pointed out that users in a FOSS community played an important role as co-developers due to the fact that a considerable use of software products by many users can lead to defects detection for the products and then the improvement of their quality. Ye et al. [24] indicated the importance of Legitimate Peripheral Participation (LPP) [14] for FOSS users. Users begin to have a sense of belonging to a community through mutual learning among the users themselves and then gradually undertake more important roles in the community. For instance, a user might only use a software product as an end-user at first, but then begin to report bugs and/or submit patches to fix the bugs through LPP.

Although the literature has been discussing the importance of users' roles and collaboration between developers and users, few studies have reported on how the informal social structure formed by developers and users affect collaboration in distributed software development. Specifically, our research questions in this paper are as follows; *What kind of roles do the key participants play in a FOSS community?* and *Who is the central participant in an infor-*

mal social structure? This paper describes our case study, which analyzes informal social structures in FOSS communities with a focus on participants who belong to both developer and user groups and assist the collaboration between them. We perform a static analysis for looking at an informal social structure in a single period and a dynamic analysis for observing the transition of the structure over time. Moreover, while our prior study [13] aimed at understanding the informal social structure in the Apache community, this paper describes a comparison study on the differences in informal social structure between the Apache community and the Netscape community.

This paper is organized as follows. Section 2 summarizes the related research. Section 3 illustrates the informal social structure formed by developers and users. Section 4 describes the analysis perspectives and used metrics used in this case study. Section 5 is our case study and Section 6 discusses the results of case study. Finally, we conclude this study in Section 7.

2. RELATED WORK

Up to now, many studies have analyzed software development practices to obtain a clear understanding of distributed software development [3, 10, 15, 21]. Mockus et al. [15] investigated some assumptions regarding FOSS development. Using CVS logs and bug report data, they have revealed that only 4% of Apache developers contributed 88% of added lines of code and 66% of fixed defects. Mockus et al. [1] investigated some assumptions regarding FOSS development. German et al. [5] also reported similar results for the Ximian project. Herbsleb et al. [6, 7] analyzed data of modification requests in a change management system and compared communication patterns in distributed development to patterns in co-located development. They have found that distributed software development took much longer and require people than that in co-located development. These studies revealed practices of distributed software development based on a quantitative analysis of the data accumulated during software development (such as change history of source code and bug reports). In this paper we also perform a quantitative analysis using the data of communication history of mailing lists and newsgroups. While the studies cited above focused on development practices of developers, we look at collaboration between developers and users, informal social structures formed by them, and key participants for the collaboration between them distributed software development.

In contrast with the quantitative analysis of distributed software development, there also exist qualitative studies on distributed software development in FOSS communities. Jensen et al. [10] illustrated role migration and advancement processes in FOSS communities including the Apache community, based on qualitative and ethnographic methods for the analyses. Nakakoji et al. [16] and Ye et al. [24] carefully observed developers' activities in FOSS communities and then modeled the motivation of FOSS developers and the evolution patterns of FOSS communities. These studies are helpful for understanding the whole picture of organizational structures in FOSS communities, but provide little knowledge of informal social structures formed by FOSS developers and users. So we analyzed the informal social structures in FOSS communities, especially focusing on coordination processes in distributed software development.

Some studies [1, 2, 8, 9] based on the Social Network Analysis (SNA) [20, 22] approach have investigated informal social structures. SNA provides both a visual (qualitative) analysis and a mathematical (quantitative) analysis of relationships among people. Hossain et al. [8] analyzed the coordination ability of employees in a large corporation using techniques of text mining and (one of metrics in SNA) Freeman's centrality [4]. Applying the techniques to email data in the corporation, they found that employees with high coordination ability had high centrality. Yamauchi et al. [23] described how distributed developers coordinated their activities and how electronic media were used in the coordination by analyzing the mailing lists of the FreeBSD Newconfig community and the GNU GCC community [23]. Howison et al. [2, 9] constructed network structures from thread trees in bug report data and analyzed relationships between bug reporters and centralities. After comparing five FOSS projects, they concluded that larger projects had key developers who kept participating in a project for substantial periods of time. We also extracted informal social structures from communication history data in online media such as mailing lists and newsgroups and we analyzed the structure based on SNA. While the studies above and our pilot study [11, 18] mainly analyzed the data from developers' mailing lists, the case study in this paper uses data collected from mailing lists and newsgroups in both a developer group and a user group in order to see coordination and collaboration processes between developers and users in FOSS communities.

3. INFORMAL SOCIAL STRUCTURE IN FOSS COMMUNITY

Since, in general, participants in FOSS communities are geographically distributed, they rely heavily on online media as a way to communicate each other. So the history of communications among participants via online media such as mailing lists, newsgroups and bulletin board systems helps us extract the informal social structure of participants in FOSS communities. In this paper, we define the informal social structure as sender-receiver relationships in online media. Figure 1 is an example of the informal social structure in a FOSS community. The left side of Figure 1 shows a thread tree in online media and the right side is a network structure constructed from the thread tree, nodes and edges representing message senders and replies respectively. For example, if participants B and C repliy to a message sent by participant A, edges are linked from nodes B and C to node A. In this way, we can represent and analyze the informal social structure in FOSS communities as network structures.

In a FOSS community, there are different types of participants (e.g., developer, tester and user), and subgroups. By using online media, technical matters (for instance, topics related to new features and/or defects of FOSS products) are discussed in a developer group, Q&As about software use is discussed in a user group, and so forth. Figure 2 illustrates an example of the informal social structure formed by two subgroups in a FOSS community. P_{dev} (yellow node) is a participant who had sent a message to the developer group (the left circle in the figure); P_{usr} (blue node) is a participant who had sent a message to the user group (the right circle in the figure); and $P_{d \cap u}$ (red node) is a participant who had sent to both the developer group and user group. By discussing topics within each subgroup, participants of

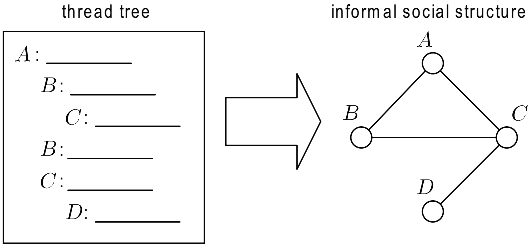


Figure 1: Construction of an informal social structure.

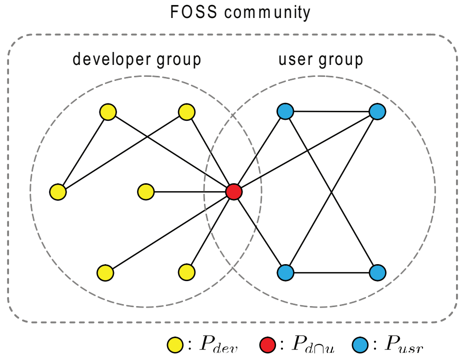


Figure 2: An example of an informal social structure in a FOSS community.

subgroups can efficiently share relevant information among themselves.

However, even if for instance, P_{usr} in the user group discussed useful ideas which might improve the quality of software products, P_{dev} in the developer group would not notice the ideas. As a result, the ideas would be buried in the communication logs of the user group.f Therefore, $P_{d\cap u}$ who participates in the both groups has the important role of transmitting feedback from users such as feature requests and/or defect reports sent to developers. In parallel, $P_{d\cap u}$ also has a role of user support because as a develoer s/he is knowledgeable about developed software as a developer. We take into consideration that $P_{d\cap u}$ who transfers information discussed in one group to other group, is largely involved in facilitating communication and collaboration among participants in the FOSS community, which means that $P_{d\cap u}$ has a strong influence on the informal social structure. In this paper, we focus on the role and network structure of $P_{d\cap u}$ for these reasons.

4. ANALYSIS METHODOLOGY

4.1 Perspective of Analysis

We can analyze the informal social structure formed by a FOSS community based on the following two perspectives on complex social networks.

4.1.1 Static Perspective

By statically capturing the informal social structure at certain points in time, we can examine the state and characteristics of the structure in detail. From this perspective, the informal social structure can be analyzed by using visual representations of the structure and metrics used in SNA. In this paper, we analyze the characteristics of the informal social structure itself and the coordination ability of each participant in the structure.

Furthermore, a FOSS community such as in Figure 2 can be seen to have three kinds of informal social structure. Analyzing $P_{d\cap u}$ in each network, we can better understand the role of $P_{d\cap u}$ as a bridge between developers and users.

- all-participants network formed by $P_{d\cap u}$, P_{dev} and P_{usr} (a FOSS community in Figure 2).
- developers network formed by $P_{d\cap u}$ and P_{dev} (a developer group in Figure 2).
- users network formed by $P_{d\cap u}$ and P_{usr} (an user group in Figure 2).

For static analysis of the informal social structure in the FOSS community, we use the average path length L and the clustering coefficient C that are widely used for measuring the complexity of network structure (L and C are described in Section 4.2).

We can also use Freeman's centralities [4] as a metric to see coordination ability. As Hossain et al. [8] reported the analysis results of the social structure in a large company based on email data, showing a significant correlation between coordination ability and Freeman's centralities (Degree, Betweenness and Closeness). Hossain et al. also found the highest correlation between the coordination ability and Betweenness Centrality, which is one of Freeman's centralities and a metric for indicating the degree of intermediation among people.

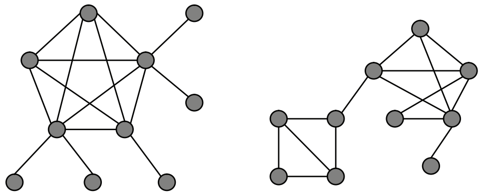
However, Howison et al. [2, 9] argued that Betweenness was not suitable for analyzing the informal social structure constructed using mailing lists data or newsgroup data, because such data are open to the public so that people can share information (i.e., a person does not need to transfer information from one person to another person). For this reason, Howison et al. used Degree Centrality for their analysis of FOSS communities. Degree Centrality is a value of a normalized degree (the number of edges) according to network size (the number of nodes). Degree Centrality is essentially the same as degree when comparing two nodes in the same network or the same scale network. For simplicity, in this paper, we use degree as a metric for the coordination ability.

4.1.2 Dynamic Perspective

Anyone can freely participate in a FOSS community and they can discuss freely and exchange ideas on an equal footing. For this reason, the informal social structure formed by participants in a FOSS community constantly changes over time. In order to analyze this *living network*, not only the static analysis described earlier, but also a dynamic, time series analysis is required. In the dynamic perspective on networks, we analyzed changes in the informal social structure and the coordination ability of $P_{d\cap u}$ over time. Since it was difficult to analyze all $P_{d\cap u}$ in large-scale networks in all period of time, we selected the top 5 $P_{d\cap u}$ those the highest degrees, and analyzed them in a time series.

4.2 Network Analysis Metrics

This section describes the average path length L and the



$L = 2.18 \quad C = 0.34$ (a)

$L = 1.84 \quad C = 0.66$ (b)

Figure 3: An example of metrics for analysis.

clustering coefficient C used in our analysis.

4.2.1 Average Path Length

The average path length L is defined as the average number of steps along the shortest paths for all reachable pairs. When the average path length of node v_i is L_i and the number of nodes is n, L is defined as follows.

$$L = \frac{1}{n}\sum_{i=1}^{n} L_i. \tag{1}$$

Generally, a network with small L is highly efficient in distributing information. However, since the informal social structure targeted in this paper is defined using data from open media such as a mailing list, it differs from general information transfer networks. Figure 3 is an example of L and C in two networks. There seem to be intensive discussions among five participants in Figure 3 (a). As a result, the value of L in Figure 3 (a) is low compared to that in Figure 3 (b). In this paper, we consider L to show not the efficiency of information transfer but the degree of concentration of discussions among participants.

4.2.2 Clustering Coefficient

The clustering coefficient C_i of node v_i is the proportion of links between two nodes which are linked to v_i. This means that C_i is the proportion of triangles among any three nodes. The clustering coefficient C of the whole network is the average number of C_i defined as follows:

$$C = \frac{1}{n}\sum_{i=1}^{n} C_i. \tag{2}$$

In Figure 3 (a), there are five participants that have only one edge (they do not from triangles with other participants). Because C_i of the participants who linked with them are low, C of the network (a) is lower than (b) as a results. Although the network is separated into two clusters in the case of Figure 3 (b), the ratio of nodes connected with each other is high and so C of the network (b) is higher.

In this paper, we consider C to show how many participants know each other. Note that, although we calculate C in developers network and users network, we do not calculate C in all-participants network. C is not suitable for applying to a network that contains $P_{d\cap u}$, P_{dev} and P_{usr}, shown in Figure 2, because P_{dev} and P_{usr} belong to different networks, that is to say, a direct edge between P_{dev} and P_{usr} (a triangle of $P_{d\cap u}$, P_{dev} and P_{usr}) does not exist in a practical sense.

Table 1: Statics of the two analysis perspective.

		Apache	Netscape
static analysis	period	2005/12	2002/08
	# participants	683	1,058
	# messages	4,068	3,509
dynamic analysis	period (from)	2001/11	1999/09
	period (to)	2007/09	2007/02
	# participants	12,710	23,386
	# messages	112,143	91,363

4.2.3 Average Number of Edges

The average number of edges $\langle k \rangle$ of each node is a fundamental metric in social network analysis. $\langle k \rangle$ is defined as follows.

$$\langle k \rangle = \frac{1}{n}\sum_{i=1}^{n} k_i. \tag{3}$$

5. CASE STUDY

This section describes our case study, in which we used communication history data collected from two well-known FOSS communities, the Apache HTTP Server community and the Netscape Browser community. We compare the informal social structures of the two communities.

5.1 Target Community

Apache HTTP Server and Netscape Browser are very popular FOSS.

- Apache HTTP Server[1]:

  As of February 2008, Apache had a dominant share, 50.6%, in the world's HTTP server market [17]. The Apache community has been developing three major versions in parallel, and they continue to release many security patches. Apache HTTP Server is widely recognized as high quality and reliable software.

- Netscape Browser[2]:

  Netscape is a Web browser software which dominated with 80% of the browser market at its peak in 1996. Influenced by a study by Raymond [19], the Netscape community decided in1998 to open its source code in order to pit Netscape against Internet Explorer. The community, however, could not win back its market share. In the end, the Netscape community announced a halt to software development and user support on March 1st, 2008.

5.2 Data Source

The history of communications in Apache and Netscape were collected from online media used for public discussions. Apache used mailing lists and Netscape used newsgroups. We considered that there is little difference between the two media for our purposes of analyzing informal social structure. The difference between mailing lists and newsgroups is in how messages are received. In the case of a mailing list, messages are automatically distributed to users' mailboxes if users subscribe to a mailing list in advance. In case

[1]http://httpd.apache.org/

[2]http://browser.netscape.com/

Table 2: Network analysis metrics when a period of each major version was released (static analysis).

		Apache ($P_{d\cap u}$)	Netscape ($P_{d\cap u}$)
all participants	# nodes	661 (24)	1,022 (38)
	# edges	1230 (720)	1,600 (221)
	$\langle k \rangle$	1.861	1.566
	L	3.339	4.576
	C	0.104	0.083
developers	# nodes	131 (24)	319 (38)
	# edges	431 (302)	375 (121)
	$\langle k \rangle$	3.290	1.176
	L	2.757	6.643
	C	0.212	0.037
users	# nodes	554 (24)	741 (38)
	# edges	861 (540)	1246 (154)
	$\langle k \rangle$	1.554	1.682
	L	3.308	3.112
	C	0.085	0.101

of newsgroups, users need to choose and receive messages by themselves. The difference has little effect on the analysis of the informal social structure, which is formed by the history of message sending. Moreover, though our analysis period coincides with the decline of newsgroups, there is no published indicating that the Netscape community shifted to other media, such as mailing lists.

Table 1 shows the number of unique participants and the number of messages for our analysis. For the static analysis, we selected the data when the major versions were released (Apache: ver.2.2.0, Netscape: ver.7.0). For the dynamic analysis, we used all the data we could collect. Since many discussions go on for more than a month, we used a time window with month m and $m \pm 1$ [9].

For instance, when we refer to the network structure in May, this means that the structure is defined by using data from April to June.

5.3 Result

5.3.1 *Results of static analysis*

For the static analysis, we used Pajek[3], an excellent tool for large network analysis. It is widely used for visualizing large-scale networks and calculating metrics. As a graphing algorithm of a network, we used the Kamada-Kawai algorithm [12] that is one of the spring layout algorithms. Note that, in the drawing of each network, we exclude nodes with no edge (i.e., a message was sent but there was no reply) in order to simplify the image since each network is very large and complex.

Figure 4 shows informal social structures in each FOSS community. The left side in Figure 4 shows Apache' networks and the right side shows Netscape. The top in Figure 4 shows the all-participants network ($P_{d\cap u}$, P_{dev} and P_{usr}), the center shows the developers network ($P_{d\cap u}$ and P_{dev}), and the bottom is the users network ($P_{d\cap u}$ and P_{usr}). Node size indicates the number of degrees (a node with many edges is drawn as a large node). Node colors shows the participant type as in Figure 2; yellow node (P_{dev}), blue node (P_{usr}) and red node ($P_{d\cap u}$). The metrics which indicate features

[3]http://vlado.fmf.uni-lj.si/pub/networks/pajek/

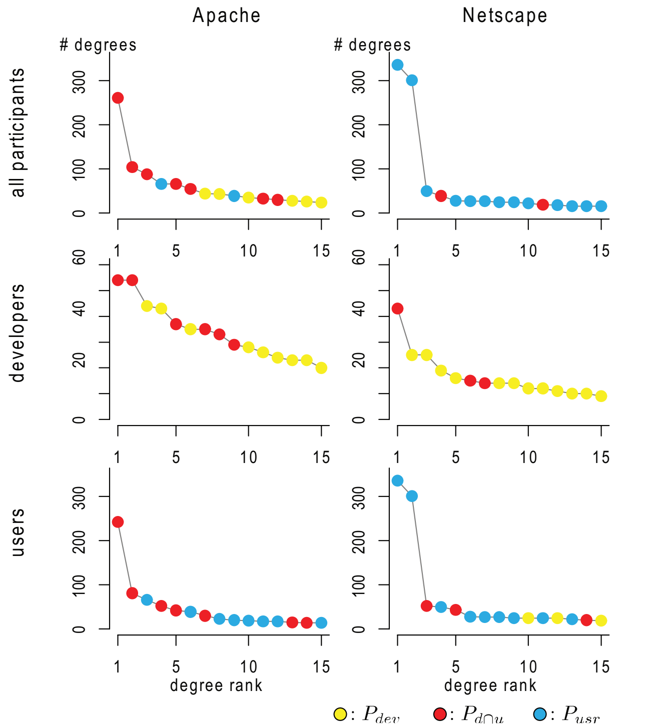


Figure 5: Degrees of each of the top 15 participants in descending order of degree (static analysis).

of each network structure are shown in Table 2. The values in parenthesis show only about $P_{d\cap u}$.

The order of degrees of each node is shown in Figure 5 to confirm the specific number of degree, though participants with high coordination ability (i.e. means has many degrees) in each network are seen in Figure 4. The x-axis and y-axis mean the number of degrees and the ranking of degree, respectively.

First, we describe the difference between Apache and Netscape regarding the all-participants network (the top side of Figure 4). The size of network (the number of nodes and edges) of Netscape is 1.5 times larger than Apache in Table 2. The number of $P_{d\cap u}$ of Netscape is also larger than Apache (Apache: 24, Netscape: 34). However, $P_{d\cap u}$ in Apache has more edges than $P_{d\cap u}$ in Netscape (Apache: 720, Netscape: 221) and has sent more messages. The results are also seen in Figure 4.

Next, we describe the difference between Apache and Netscape regarding the developer ($P_{d\cap u}$ and P_{dev}) network (the center of Figure 4). Although Table 2 shows that the number of developers of Apache is half that of Netscape, Apache developers have more edges, the average number of edges $\langle k \rangle$ being more than twice that of large as Netscape (Apache: 302, Netscape: 121). The average path length L of the Apache developers network is the smallest of all networks and the clustering coefficient C is the largest due to the large $\langle k \rangle$. On the other hand in Netscape, L is the largest and C is the smallest of all networks. In the Apache developers network (left center in Figure 4), we seek a high density of many edges and small nodes with few edges at marginal places. The network seems rather like a star network in broad perspective. Meanwhile, the Netscape developers network is

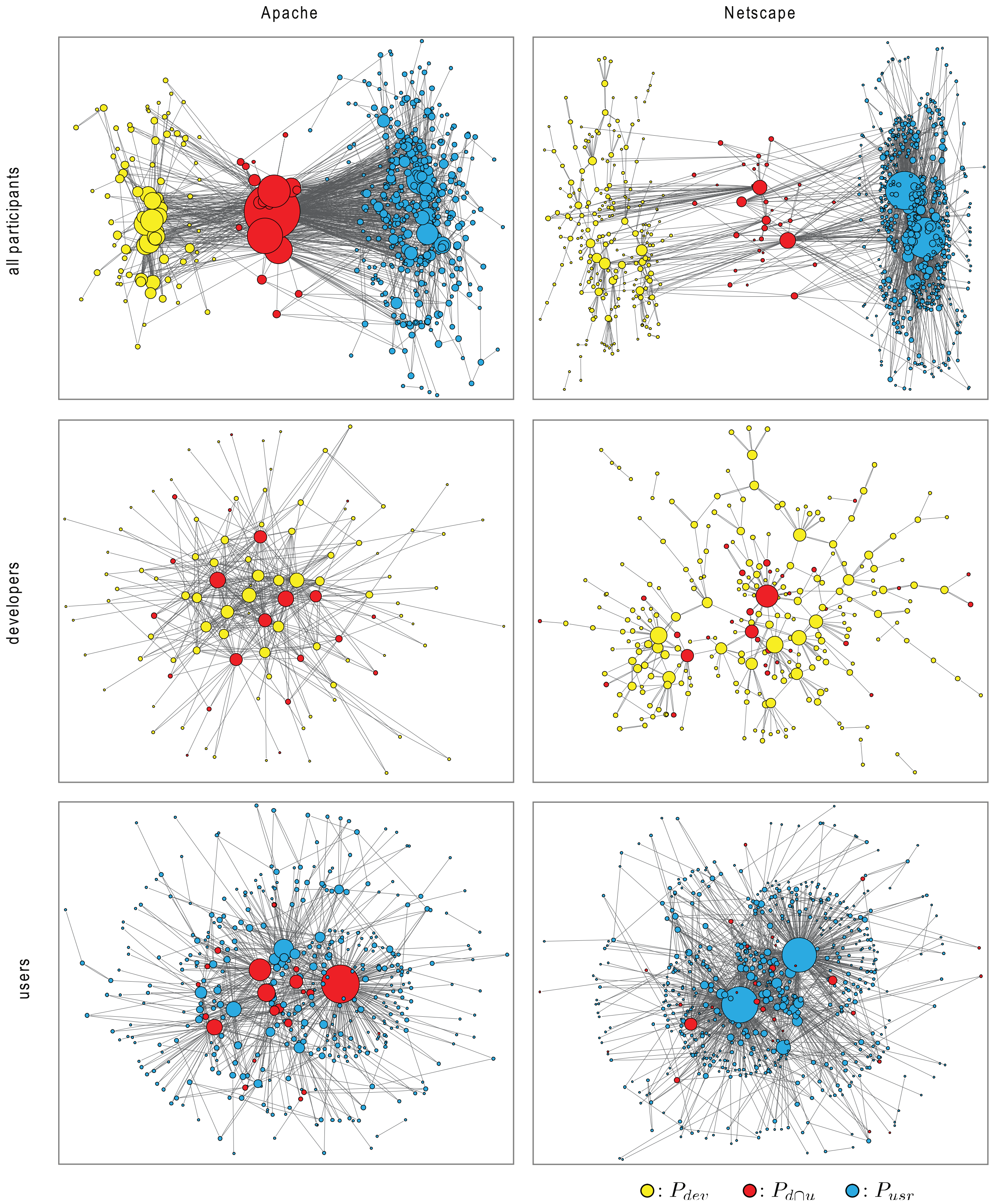


Figure 4: Informal social structures in Apache and Netscape (static analysis).

formed by many little stars with one big node and about ten little nodes. In other words, the structures of developers networks in Apache and Netscape differ greatly. Figure 5 also illustrates this difference. Apache and Netscape, respectively, have six and three $P_{d\cap u}$ of their top 15 participants with high degrees.

Finally, we focus on the users ($P_{d\cap u}$ and P_{usr}) network (the bottom of Figure 4). The difference between the structure of the users network in Apache and Netscape cannot be confirmed in Table 2, because there is little difference in $\langle k\rangle$, L and C. However, in Figure 4, the Apache users network has a structure where the highest degree $P_{d\cap u}$ is centered and some high degree $P_{d\cap u}$ form a huge cluster. In contrast, the Netscape users network has a structure with two large clusters centered around two high degree P_{usr}. These results are also shown in Figure 5. Moreover, there are only two $P_{d\cap u}$ with high degrees among the top 15 nodes in the Netscape users network. In other words, the users networks of Apache and Netscape differ in the type of their central participants (Apache: $P_{d\cap u}$, Netscape: P_{usr}) rather than in structure.

5.3.2 Results of dynamic analysis

In the dynamic perspective on networks, we focused on the transition of informal social structures over time. Figure 6 shows the changes in metrics of the all-participants network over time. The left side shows Apache and right side shows Netscape. At the top is the number of edges and nodes and at the bottom the $\langle k\rangle$ and L. The vertical dashed line indicates the period of the static analysis (major version released).

In case of Apache, the number of nodes reached a peak around 2002 and has been gradually decreasing since. However, the number of edges, L and C changed little over the whole period. In other words, the Apache network is relatively stable in the size and structure. In the static analysis, we also saw its stability.

In contrast, the number of nodes and edges in the Netscape network decreased over the whole period. Since $\langle k\rangle$ gradually increased especially from 2000 to 2004, nodes with many degrees remained instead nodes have few edges decreased. In the static analysis, we analyzed the period during with the size of the Netscape network was decreasing.

Next, we describe changes in the ranking over time of the top five $P_{d\cap u}$ with many degrees. Figure 7 shows changes of the ranking of $PN_{d\cap u}(N = \{1, 2, ..., 5\})$. $PN_{d\cap u}$ means the top N of high degree $P_{d\cap u}$ in the period of the static analysis.

In Figure 7, $P1_{d\cap u}$ in the Apache community is on top for the whole period (for about six years). The top two and three are occupied by $P2_{d\cap u}$, $P3_{d\cap u}$, $P4_{d\cap u}$ and $P5_{d\cap u}$ after 2004. The static analysis suggesting that the network structure in Apache is stable was confirmed here. Still the result of the dynamic analysis indicates that a $P_{d\cap u}$ who links two networks (developers and users) and has a high communication ability is also shows little change over time in the Apache community.

In contrast, $P1_{d\cap u}$ of the Netscape community maintains a high ranking only around the period 2002-2004. $P2_{d\cap u}$ frequently moves in the ranking and seems to fluctuate, sending many messages or few in different months. $P3_{d\cap u}$, $P4_{d\cap u}$ and $P5_{d\cap u}$ are shown in the graph for only five months. This means there is no $P_{d\cap u}$ that maintains a high communication ability over the time while the scale of the Netscape community decreased.

6. DISCUSSIONS

6.1 Lessons learned

From the results of the case study, we have found the differences between Apache and Netscape; 1) the developers networks differ in their structures, 2) users networks differ in the type of their central participants, 3) the all-participants networks differ in the lifetime of $P_{d\cap u}$ with a high communication ability. This section discusses these three results.

The Apache developers network had high C. The density of about 30 developers, in particular, was very high. It can be seen that highly motivated developers discuss actively with each other. The Netscape developers network had a Q&A structure rather than a discussion structure; that is to say that the structure was constructed by nodes with high degree connected to many nodes with low degrees. The result would indicate that discussions in Netscape did not concentrate at one point and that, therefore, collaboration among Netscape developers was difficult because there was no $P_{d\cap u}$ with a high communication ability.

In the Apache users network, the number of $P_{d\cap u}$ who answers users questions with high degrees was large, and the network had a structure that allows $P_{d\cap u}$ to absorbs users' comments and ideas and transmits them to the developer group smoothly. The users network in Netscape had few $P_{d\cap u}$ and the structure formed around two P_{usr} with high degree. These results indicate that the Apache network enjoyed more collaboration among participants.

In the case of the all-participants network, $P_{d\cap u}$ in the Apache network communicated more actively with P_{dev} and P_{usr} than did $P_{d\cap u}$ in Netscape. Developers and users in Apach seem to have a better relationship. Additionally there was $P1_{d\cap u}$ who has always showed the highest degree seeming to work as a powerful leader with high motivation. The results also live with a study by Howison et al. [9] that analyzed developers' communities.

We conclude that the key factors for the success in geographically distributed software development such as a FOSS community are the existence of highly motivated participants like Linus Torvalds in the Linux community and a $P_{d\cap u}$ who helps developers and users collaborate closely with each other.

6.2 Threats to Validity

In this paper, we regarded as message senders/receivers those whose email addresses corresponded exactly in the communication history data as a same person. However, we also confirmed that some persons used multiple email addresses or changed their email addresses. In the near future we need to analyze more cleaned data by using, for example, the email address processing method proposed by Bird et al. [1] which uses characters before "@" and then considers first name or last name based on a clustering technique.

We treated an informal social structure as an undirected graph in our analysis, but we can define directed relationships between message sender/receivers. We attempted to analysis in consideration of direct edges as in the research of Bird et al. [1].

In this paper, we compared FOSS communities that develop different types of software products. Although the

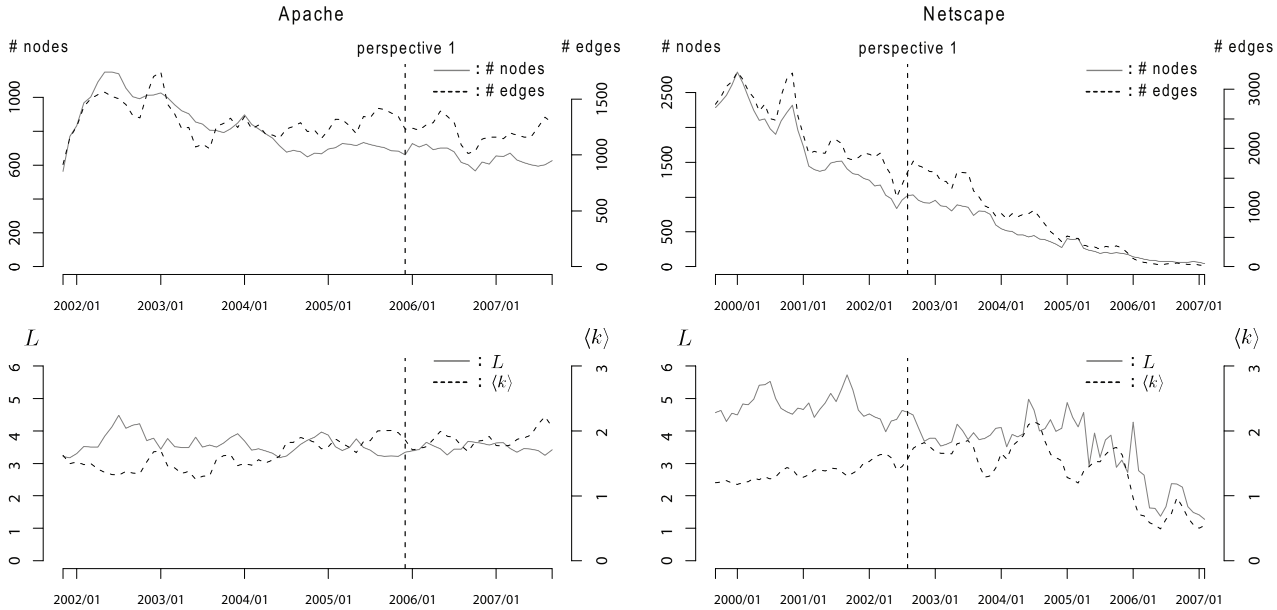


Figure 6: Changes in the four network analysis metrics (dynamic analysis).

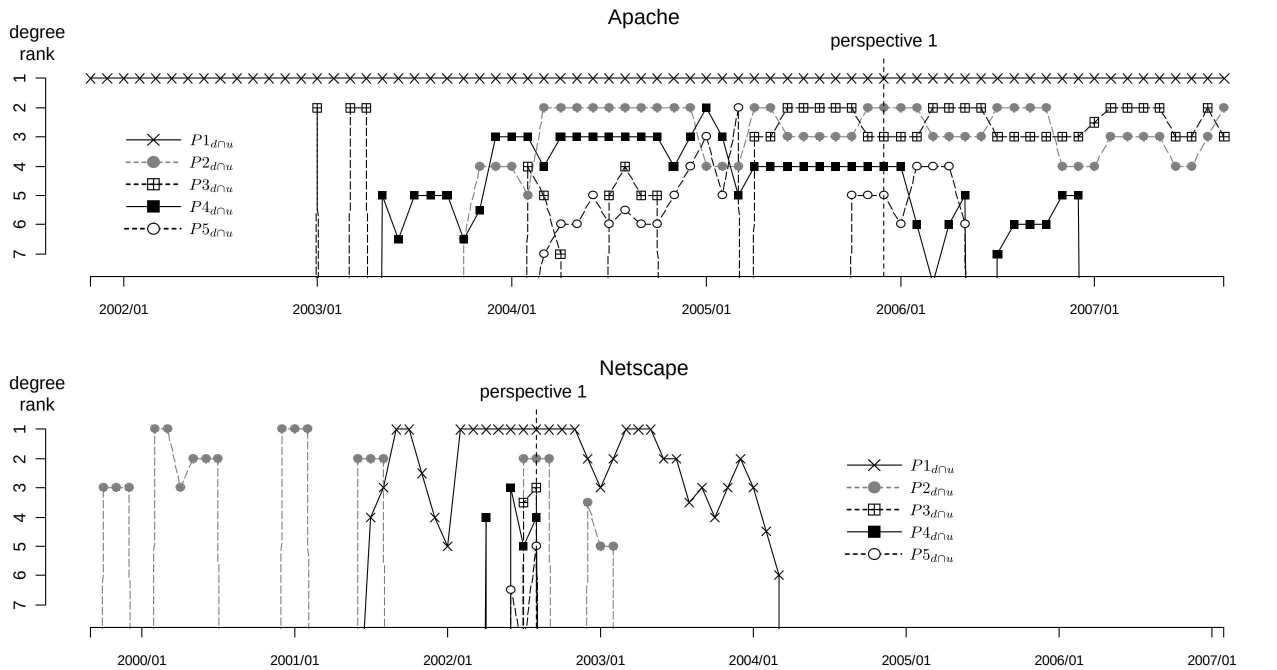


Figure 7: Changes in degree ranking of the five $P_{d\cap u}$s when the major version was released (dynamic analysis).

Apache community develops HTTP server software, and many of its users would have some expertise (e.g. as administrators of HTTP servers). Netscape is Web browser software and its users would range from beginners to experts. The difference in types of users could affect the analysis of $P_{d\cap u}$, but we believe that the results of our analysis can contribute to identifying partially factors of success and failure of FOSS communities and distributed software development, from a perspective different from the study by Mockus et al. [15].

Moreover, although we focused on $P_{d\cap u}$ who had participated in both a developer group and a user group, we are not clear about whether $P_{d\cap u}$ actually coordinate between the two groups or not. In our future work, we plan to analyze coordination activities of $P_{d\cap u}$ by applying a natural language processing technique to bodies of messages sent by $P_{d\cap u}$.

7. CONCLUSION AND FUTURE WORK

In this paper, we analyzed informal social structures with a focus on participants who belong to developers and users groups and who assist in the collaboration between them. Our research questions in this paper were as follows; *What kind of roles does the key participant play in a FOSS community?* and *Who is the central participant in an informal social structure?* Our findings are the following.

- In case of Apache, there was a single participant who always has the highest degree. S/he seems to work as a powerful leader with high motivation.
- There is no participant who has a high communication ability in Netscape. Therefore discussions in Netscape do not concentrate at one particular point and then collaboration among Netscape developers is difficult. The structure also hinders not good for collaboration between developers and users.
- The key factors for the success in distributed software development such as a FOSS community are, is the contribution of highly motivated participants who help developers and users collaborate closely with each other.

Although we used degrees as a metric for the coordination ability in this paper, the meaning of message sending to a developer group is semantically different from message sending to a user group. In the future, we will propose metrics that represent the coordination ability of the participants to bridge the two groups.

8. ACKNOWLEDGMENTS

This research is being conducted as a part of the Next-generation IT program and Grant-in-aid for Scientific Research (B) 17300007, 2007 and for Young Scientists (B), 17700111, 2007 and 20700028, 2008, by the Ministry of Education, Culture, Sports, Science and Technology.

9. REFERENCES

[1] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan. Mining email social networks. In *Proc. the 2006 International Workshop on Mining Software Repositories (MSR'06)*, pages 137–143, 2006.

[2] K. Crowston and J. Howison. The social structure of free and open source software development. *First Monday*, 10(2), 2005.

[3] J. Feller and B. Fitzgerald. *Understanding Open Source Software Development*. Addison-Wesley, 2002.

[4] L. C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1(3):215–239, 1979.

[5] D. German and A. Mockus. Automating the measurement of open source projects. In *Proc. the 3rd Workshop on Open Source Software Engineering*, pages 63–67, 2003.

[6] J. D. Herbsleb and A. Mockus. An empirical study of speed and communication in globally distributed software development. *IEEE Trans. Software Engineering (TSE)*, 29(6):481–494, 2003.

[7] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter. An empirical study of global software development: Distance and speed. In *Proc. the 23rd International Conference on Software Engineering (ICSE'01)*, pages 81–90, 2001.

[8] L. Hossain, A. Wu, and K. K. S. Chung. Actor centrality correlates to project based coordination. In *Proc. the 20th Conference on Computer Supported Cooperative Work (CSCW'06)*, pages 363–372, 2006.

[9] J. Howison, K. Inoue, and K. Crowston. Social dynamics of free and open source team communications. In *Proc. the 2nd International Conference on Open Source Systems (OSS'06)*, pages 319–330, 2006.

[10] C. Jensen and W. Scacchi. Role migration and advancement processes in ossd projects: A comparative case study. In *Proc. the 29th International Conference on Software Engineering (ICSE'07)*, pages 364–374, 2007.

[11] T. Kakimoto, Y. Kamei, M. Ohira, and K. Matsumoto. Social network analysis on communications for knowledge collaboration in oss communities. In *Proc. the 2nd International Workshop on Supporting Knowledge Collaboration in Software Development (KCSD'06)*, pages 35–41, 2006. Tokyo, Japan.

[12] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

[13] Y. Kamei, S. Matsumoto, H. Maeshima, Y. Onishi, M. Ohira, and K. Matsumoto. Analysis of coordination between developers and users in the apache community. In *Proc. the 4th International Conference on Open Source Systems (OSS'08)*, Milan, Italy, to appear.

[14] J. Lave and E. Wenger. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, 1991.

[15] A. Mockus, R. T. Fielding, and J. D. Herbsleb. Two case studies of open source software development: Apache and mozilla. *ACM Trans. Software Engineering and Methodology*, 11(3):309–346, 2002.

[16] K. Nakakoji, Y. Yamamoto, Y. Nishinaka, K. Kishida, and Y. Ye. Evolution patterns of open-source software systems and communities. In *Proc. the 5th International Workshop on Principles of Software Evolution (IWPSE'02)*, pages 76–85, 2002.

[17] Netcraft Ltd. Netcraft web server survey. available from ⟨http://news.netcraft.com/⟩, accessed 2008-03-04.

[18] M. Ohira, T. Ohoka, T. Kakimoto, N. Ohsugi, and K. Matsumoto. Supporting knowledge collaboration using social networks in a large-scale online community of software development projects. In *Proc. the 12th Asia-Pacific Software Engineering Conference (APSEC'05)*, pages 835–840, 2005.

[19] E. S. Raymond. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly and Associates, 1999.

[20] J. Scott. *Social Network Analysis: A Handbook*. SAGE Publications, 2000.

[21] S. K. Sowe, I. G. Stamelos, and I. M. Samoladas. *Emerging Free and Open Source Software Practices*. IGI Publishing, 2007.

[22] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[23] Y. Yamauchi, M. Yokozawa, T. Shinohara, and T. Ishida. Collaboration with lean media: How open-source software succeeds. In *Proc. the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW'00)*, pages 329–338, 2000.

[24] Y. Ye and K. Kishida. Toward an understanding of the motivation of open source software developers. In *Proc. the 25th International Conference on Software Engineering (ICSE'03)*, pages 419–429, 2003.