

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 of FOSS communities already have tremendous success in the 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 developers and users groups and assist the collaboration between them. In the case study, we perform a static analysis for looking at an informal social structure at a single period and a dynamic analysis for observing the transition of the structure over time. As the results of the case study, 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 closely collaborate 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 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 have 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 code [1]. The analysis results indicated that developers communicating with many other developers more contributed 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 a FOSS development. Raymond et al. [19] pointed out that users in a FOSS community played an important role as co-developers. The argument has been derived from the fact that a considerable use of software products by many users can lead defects detection for the products and then the improvement of the quality of the products. 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 users and then gradually undertake more important roles in the community. For instance, at the onset a user may only use a software product as an end-user, but s/he might begin to report bugs and/or submit patches to fix the bugs through LPP.

Although the literature has been discussing the importance of user's roles and collaboration between developers and users, few study has reported on how the informal social structure formed by developers and users have effect on collaboration in distributed software development. Specifically, our research questions in this paper are as follows; *What kind of roles does the key participant play in FOSS*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

community? and *Who is the central participant in the informal 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 developers and users groups and assist the collaboration between them. In the case study, we perform a static analysis for looking at an informal social structure at a single period and a dynamic analysis for observing the transition of the structure over time. Moreover, while our prior study [13] only have tried to understand the informal social structure in the Apache community, this paper describes a comparison study on the differences of the informal social structure between the Apache community and the Netscape community.

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

2. RELATED WORK

So far, many studies have analyzed software development practices to obtain clear understandings of distributed software development [3, 10, 15, 21].

Mockus et al. [15] investigated some assumptions on FOSS development. Using CVS logs and bug reports data, they have revealed that only 4% of Apache developers contributed to 88% of added lines of code and 66% of fixed defects. German et al. [5] also has reported similar results for the Ximian project. Herbsleb et al. [6, 7] has analyzed data of modification requests in a change management system and compared communication patterns in distributed development to that 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 above have revealed some practices of distributed software development based on the 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 the quantitative analysis using the data of communication history of mailing list and newsgroup. While the previous studies above focused on development practices of developers, our case study focuses on 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] have deeply observed developer's activities in FOSS communities and then modeled the motivation of FOSS developers and the evolution patterns of FOSS communities invoked by the motivated developers. 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 analyze the informal social structures in FOSS com-

munities, especially focusing on coordination processes in distributed software development.

Some previous studies [1, 2, 8, 9] considered the informal social structures based on Social Network Analysis approach (SNA) [20, 22] which 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 Freeman's centrality [4] (is one of metrics in SNA). Applying the techniques to email data in the corporation, they have 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 reports data and analyzed relationships between bug reporters and centralities. As the results of their case study of 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 extract informal social structures from communication history data in online media such as mailing list and newsgroup and analyze the structure based on SNA. While the studies above and our pilot study [11, 18] mainly analyzed the data from developer's mailing lists, the case study in this paper uses data collected from mailing lists and newsgroups in both a developers group and a users 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, and they rely heavily on online media as a way to communicate each other. So the history of communications among participants on 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 by sender-receiver relationships on online media. Figure 1 is an example of the informal social structure in FOSS community. The left side in Figure 1 shows a thread tree in online media and the right side is a network structure constructed from the thread tree, regarding nodes and edges as message senders and replies respectively. For example, if a participant B and C replies to a message sent by participant A , edges are linked from node 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 FOSS community, there are some types of participants (e.g. developer, tester and user), and subgroups according to the types of participants. Through using online media, technical matters (for instance, topics related to new features and/or defects of FOSS products) are discussed in a developers group, Q&As about usage of software are discussed in users group, and so forth. Figure 2 illustrates an example of the informal social structure formed by two subgroups in FOSS community. P_{dev} (yellow node) is a participant who had sent a message to the developers group (the left circle in

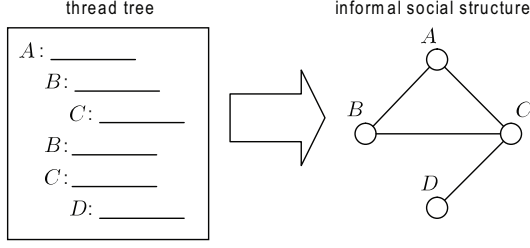


Figure 1: Construction of an informal social structure.

the figure), P_{usr} (blue node) is a participant who had sent a message to the users group (the right circle in the figure), and $P_{d\cap u}$ (red node) is a participant who had sent to both the developers group and users group. By discussing topics corresponding to the types of each subgroup, participants of subgroups can efficiently obtain relevant information for themselves.

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

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 some points in time, we can examine the state and characteristics of the structure in detail. In this perspective, the informal social structure can be analyzed by using visual representations of the structure and previously proposed metrics. In this paper, we analyze characteristics of the informal social structure itself and the coordination ability of each participant in the structure.

Furthermore, a FOSS community such as 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 $P_{d\cap u}$ which bridges between developers and users.

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

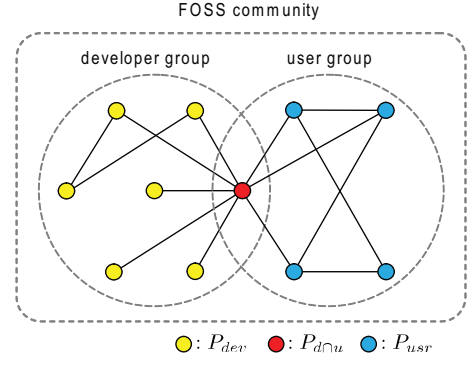


Figure 2: An example of an informal social structure in a FOSS community.

- developers network formed by $P_{d\cap u}$ and P_{dev} (a developers group in Figure 2).
- users network formed by $P_{d\cap u}$ and P_{usr} (a users 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 feature of complex network structure (L and C are described in Section 4.2).

We can also use Freeman's centralities [4] as a metric to see the ability of coordination. As Hossain et al. [8] reported the analysis results of the social structure in a large company based on email data, there is 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 inter-mediation 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 the data are open to 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 normalized degree (the number of edges) by network size (the number of nodes). Degree Centrality is essentially same as degree when comparing two nodes in a same network or same scale of 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 freely discuss and exchange ideas on equal footing. For this reason, the informal social structure formed by participant in the FOSS community constantly changes over time. In order to analyze this *living network*, not only the static analysis described earlier but also the dynamic, time series analysis is required. In the dynamic perspective on networks, we analyze changes of 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

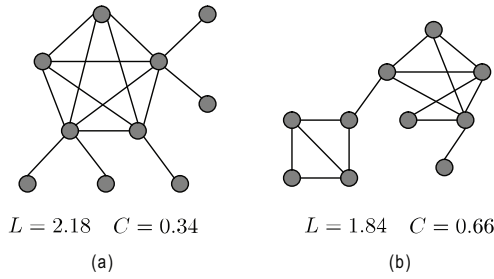


Figure 3: An example of metrics for analysis.

all period of time, we selected top 5 $P_{d \cap u}$ with the highest degrees and analyzed them in time series.

4.2 Network Analysis Metrics

This section describes the average path length L and the clustering coefficient C which are 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 of nodes. When the average path length of node v_i is L_i and the number of nodes is n , L is defined as follows.

$$L = \frac{1}{n} \sum_{i=1}^n L_i. \quad (1)$$

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 mailing list, it differs from general information transfer networks. Figure 3 is an example of L and C of two networks. There seem to be intensive discussions among five participants while 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 shows 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 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.

$$C = \frac{1}{n} \sum_{i=1}^n C_i. \quad (2)$$

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

In this paper, we consider C shows how many participants know each other. Note that, although we calculate C to the developers network and the users network, we do not calculate C to the all participants network. C is not suitable

Table 1: Statics for the two analysis perspective.

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

for applied to a network that contains $P_{d \cap u}$, P_{dev} and P_{usr} as shown in Figure 2, because P_{dev} and P_{usr} belong to a different network, that is, 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.

5. CASE STUDY

This section describes our case study in which we used communication history data collected from two well-known FOSS communities, Apache HTTP Server community and 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 Server¹:

In February 2008, Apache has a dominant share of 50.6% in the world's HTTP server market [17]. 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 reliability software.

• Netscape Browser²:

Netscape is a Web browser software which dominated 80% of the browser market at peak in 1996. Although influenced by study by Raymond [19], the Netscape community decided to open its source code in order to pit against Internet Explorer in 1998, the community could not claw back the market share. The Netscape community lastly announced the end of 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 that was used for public discussions. Apache used mailing lists and Netscape used newsgroups. We consider that there is little difference between the two media for our analysis of the informal social structure. The difference between mailing lists and newsgroups is how to receive messages. In case of mailing list, messages are automatically distributed to users' mailbox if users subscribe a mailing list in advance. In case of newsgroups, users need to choose and receive messages by themselves. So, the difference between discussion media has little effect on the analysis of the informal social structure which is

¹<http://httpd.apache.org/>

²<http://browser.netscape.com/>

Table 2: Results of all metrics.

		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
developers	C	0.104	0.083
	# nodes	131 (24)	319 (38)
	# edges	431 (302)	375 (121)
	$\langle k \rangle$	3.290	1.176
users	L	2.757	6.643
	C	0.212	0.037
	# 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

formed by the history of actions of message sending. Moreover, though the analysis period coincides with the decline phase of newsgroups itself, there are no publications that 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 select the data when the major version was released (Apache: ver.2.2.0, Netscape: ver.7.0). For the dynamic analysis, we use all the data we could collect. Since there are many discussions crossing over months, we use the time window with month m and $m \pm 1$ [9]. For instance, when we mention the network structure in May, it means 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 Pajek³ that is 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 spring layout algorithms. Note that, in each drawn network we excluded nodes with no edge (i.e., s/he sent a message but there was no reply) for improving the visibility because 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 is Netscape. The top in Figure 4 shows all participants network ($P_{d \cap u}$, P_{dev} and P_{usr}), the center shows developers network ($P_{d \cap u}$ and P_{dev}), and the bottom is users network ($P_{d \cap u}$ and P_{usr}). The size of nodes means the number of degrees (a node with many edges is drawn as a large node). The colors of nodes mean participant type as same as Figure 2; yellow node (P_{dev}), blue node (P_{usr}) and red node ($P_{d \cap u}$). The metrics which indicate features of each network structure are shown in Table 2. The values in parenthesis show only about $P_{d \cap u}$.

Though participants who has the high coordination ability (means has many degrees) in each network are visible in

³<http://vlado.fmf.uni-lj.si/pub/networks/pajek/>

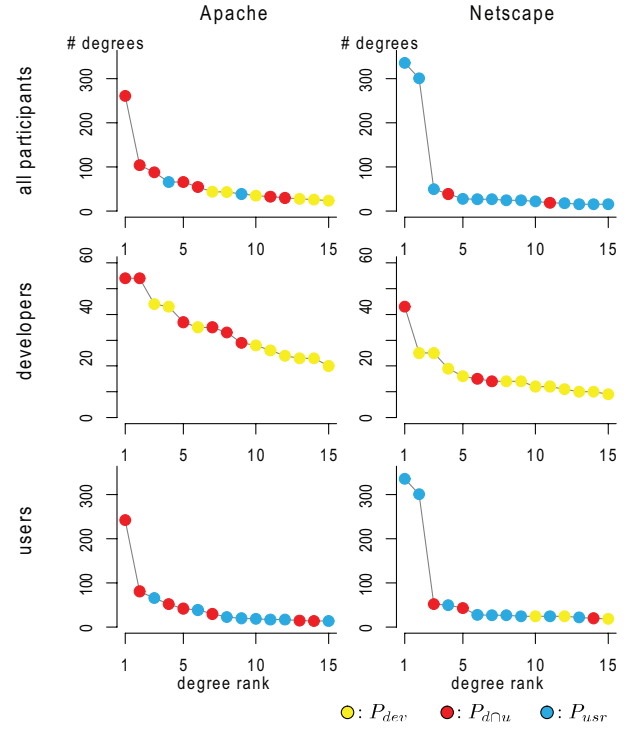


Figure 5: Degrees of top 15 participants in Apache and Netscape.

Figure 4, the order of degrees of each node are shown in Figure 5 for confirming the specific number of degree. The x-axis and y-axis respectively mean the number of degrees and the ranking of degree.

First, we describe the difference of the all participants network between Apache and Netscape (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 sent many messages. The result is also visible in Figure 4.

Next, we describe the difference of the developer ($P_{d \cap u}$ and P_{dev}) network between Apache and Netscape (the center of Figure 4). Although Table 2 shows the number of developers of Apache is half of developers of Netscape, Apache developers have many edges and the average number of edges $\langle k \rangle$ is more than twice as large as Netscape (Apache: 302, Netscape: 121). The average path length L of the Apache developers network is the smallest of all other 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 the other networks. In the Apache developers network (the left center Figure 4), many big nodes with many edges are highly dense and small nodes with few edges are placed at marginal places. The network looks like a star network in broad perspective. Meanwhile, the Netscape developers network is formed by many little stars which includes one big node and about little 10 nodes. In other words, the structure of developers networks is very different between Apache and Netscape. Figure 5 also complements

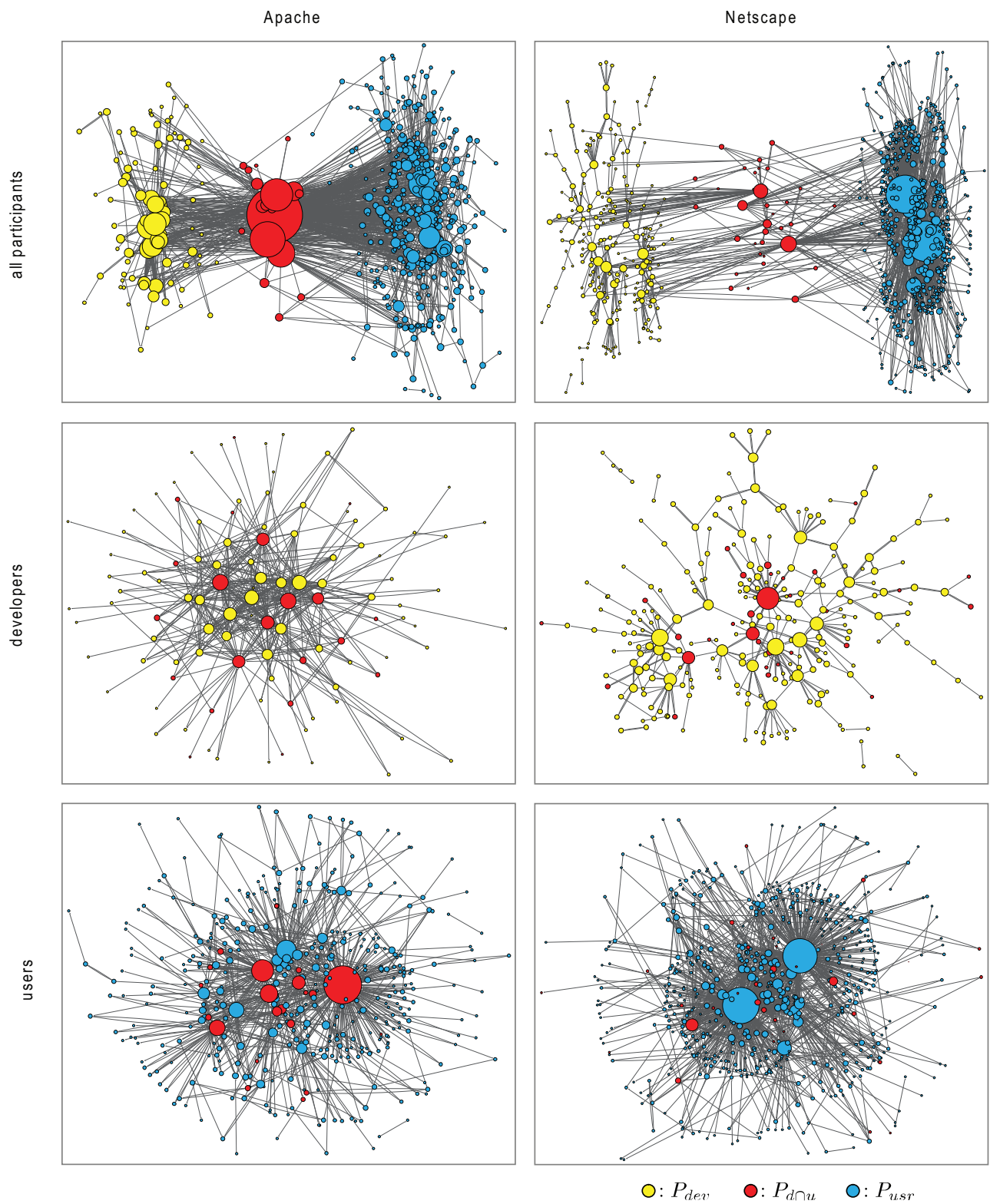


Figure 4: Informal social structures in Apache and Netscape.

the fact of the difference of the developers network between Apache and Netscape. Apache and Netscape respectively have six and three $P_{d \cap u}$ of the 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 of the structure of the users network between 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 the structure where the highest degree $P_{d \cap u}$ centered and some high degree $P_{d \cap u}$ form a huge cluster. In contrast, the Netscape users network has the structure with two large clusters centered around two high degree P_{usr} . These results are also shown in Figure 5. Moreover, the number of $P_{d \cap u}$ is only two of top 15 nodes with high degrees in the Netscape users network. In other words, the users networks of Apache and Netscape differ in the type of central participants (Apache: $P_{d \cap u}$, Netscape: P_{usr}) rather than the structure.

5.3.2 Results of dynamic analysis

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

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

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

Next, we describe changes of the ranking of top five $P_{d \cap u}$ with many degrees over time. Figure 7 shows changes of the ranking of $PN_{d \cap u}(N = \{1, 2, \dots, 5\})$. $PN_{d \cap u}$ means 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 always on top in 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. While the static analysis suggested the network in Apache has the stable network structure was confirmed, the result of the dynamic analysis indicates that $P_{d \cap u}$ who links two networks (developers and users) and has the high communication ability are also little changed over time in the Apache community.

In contrast, $P1_{d \cap u}$ of Netscape community keeps high ranks only around from 2002 to 2004. $P2_{d \cap u}$ frequently moves in the ranking and seems to send many messages or few in some months. $P3_{d \cap u}$, $P4_{d \cap u}$ and $P5_{d \cap u}$ are shown in the graph only for five months. These result means there is no $P_{d \cap u}$ who has the high communication ability over time while decreasing the scale of Netscape community.

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 the structures themselves, 2) users networks differ in the type of central participants, 3) the all participants networks differ in the lifetime of $P_{d \cap u}$ with has the high communication ability. This section discusses these three results.

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

In the Apache users network, the ratio of $P_{d \cap u}$ with high degrees who answers users questions were high, and the network had the structure that allows $P_{d \cap u}$ to absorb the users' comments and ideas and transform them to the developers group smoothly. The users network in Netscape had few $P_{d \cap u}$ and ad the structure formed around two high degree P_{usr} . These results indicate the Apache network has an advantage in collaboration among participants.

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

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

6.2 Threats of Validity

In this paper, we regarded message senders/receivers whose email address completely corresponds 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 precisely cleaned data, for instance, using 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 would like to perform our analysis in consideration of direct edges like research of Bird et al. [1].

In this paper, we compared FOSS communities that develop different types of software products. Although the Apache community develops HTTP server software and many of the users would have expertise to some extent such as a administrator of HTTP servers, Netscape is Web browser software and the users would be diverse from beginners to

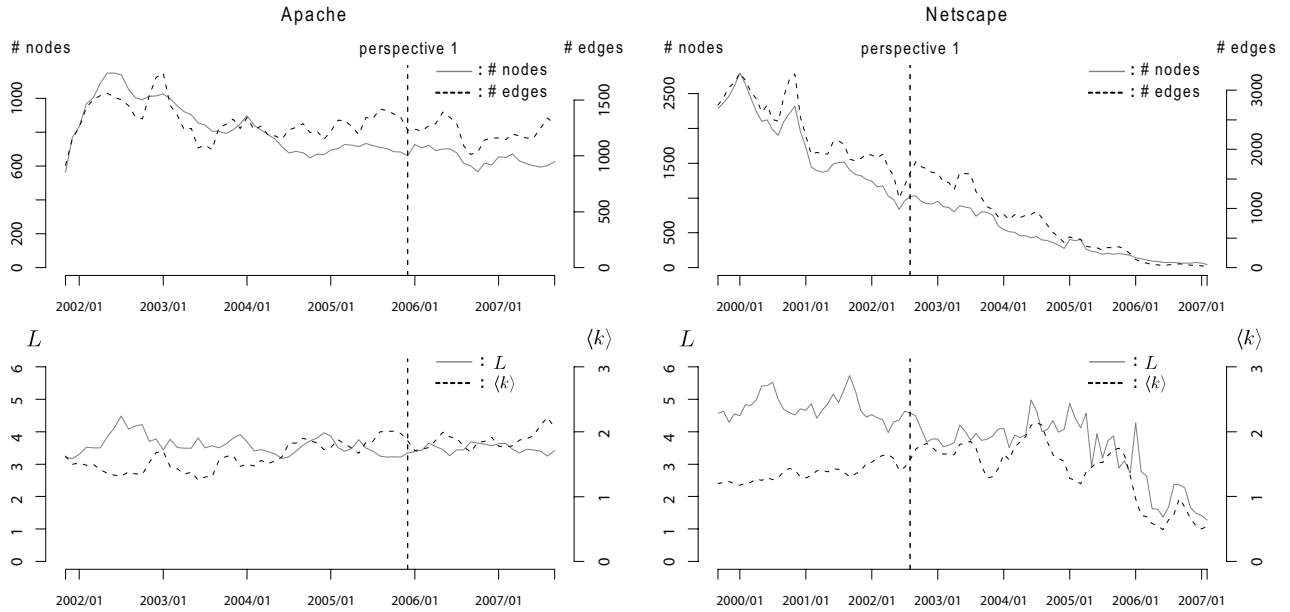


Figure 6: Changes of network metrics.

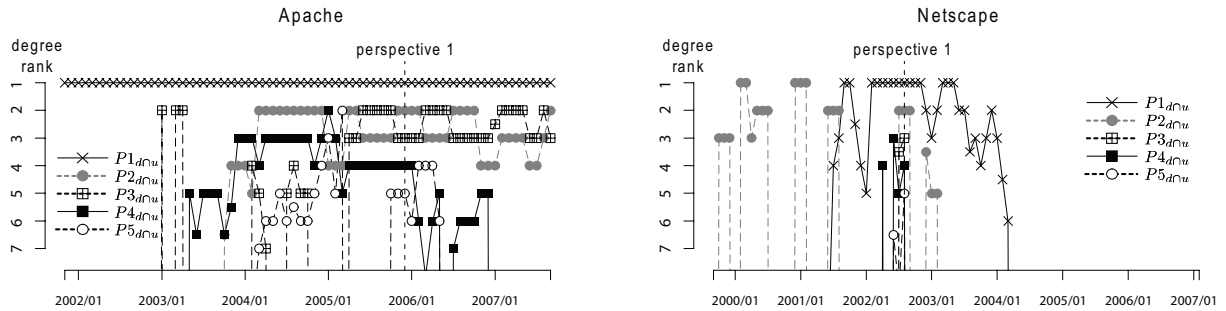


Figure 7: Changes of individual degree ranks.

experts. The difference in types of users could affect the analysis of P_{dnu} , but we believe that the results of our analysis can contribute to partially identify the factor of the success and failure of FOSS communities and distributed software development, from another perspective which is different from the study by Mockus et al. [15].

7. CONCLUSION AND FUTURE WORK

In this paper, we analyzed informal social structures with a focus on participants who belong to developers and users group and assist collaboration between them. Our research questions in this paper are as follows; *What kind of roles does the key participant play in FOSS community?* and *Who is the central participant in the informal social structure?* The findings of the case study are 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 the high communication ability in Netscape. Therefore discussions in

Netscape do not concentrate at a point and then collaboration among Netscape developers is difficult. The structure is also not good for collaboration between developers and users.

- The key factors of the success in distributed software development such as a FOSS community are the existence of highly motivated participants who help developers and users closely collaborate with each other.

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

8. ACKNOWLEDGEMENTS

This research is being conducted as a part of the EASE project in Comprehensive Development of e-Society Foundation Software program, and Stage Project, the Development

of Next Generation IT Infrastructure, and Grant-in-aid for Scientific Research (B) 17300007, 2007 and for Young Scientists (B), 17700111, 2007, by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

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 (KCS'D'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.