8-2006

Social network dynamics for open source software projects

Y. LONG

Keng SIAU
Singapore Management University, klsiau@smu.edu.sg

Citation

LONG, Y. and SIAU, Keng. Social network dynamics for open source software projects. (2006). *Proceedings of the Twelfth Americas Conference on Information Systems (AMCIS 2006), Acapulco, Mexico, August 4-6*. 820-830.
Available at: https://ink.library.smu.edu.sg/sis_research/9422

Social Network Dynamics for Open Source Software Projects

Yuan Long
University of Nebraska-Lincoln
ylong@unlserve.unl.edu

Keng Siau
University of Nebraska-Lincoln
ksiau@unlnotes.unl.edu

ABSTRACT

Drawing on social network theories and previous studies, this research is an initial effort to explore the dynamics of the social network structure in Open Source Software (OSS) teams. Three projects were selected from SourceForge.net in term of their similarities as well as their differences. Monthly data were extracted from the bug tracking system in order to achieve a longitudinal view of the interaction pattern of each project. Social network analysis was used to generate the indices of network structure. The finding suggests that the interaction pattern of OSS projects evolves from a single hub at the beginning to a core/periphery model as the projects move forward.

Keywords

Social network, social structure, Open Source Software (OSS), longitudinal study

INTRODUCTION

The phenomenon of Open Source Software (OSS) development has attracted increasing attention from both practitioners and researchers. Unlike those of the conventional software projects, the participants of OSS projects are volunteers. They are self-selected based on their interests and capability to contribute to the projects (Raymond, 2000). In addition, the developers of OSS projects are distributed all around the world. They communicate and collaborate with each other through Internet using emails or discussion boards. Therefore, effective and efficient communication and collaboration are critical to OSS success. However, little empirical research has been conducted to explore the interaction pattern of OSS teams. To fill this gap, this study aims to explore the evolution of social network structure in OSS teams. The study contributes to the enhancement of the understanding of OSS development, and provides foundation for future studies to explore the causes and effects of social networks in the OSS context.

The reminder of the paper is structured as follows. First, a literature review is provided to introduce the current studies concerning social network structure in OSS teams. Second, the theoretical foundation including social structure and social network theory is discussed. Third, the research methodology is presented, and the research results are reported. Discussions of the results, the limitations, and the implications are provided subsequently. The paper concludes with suggestions for future research.

LITERATURE REVIEW

The phenomenon of OSS development has attracted considerable attention from both practitioners and researchers in diverse fields such as computer science, social psychology, organization, and management. Because of the multifaceted nature of OSS, researchers have investigated OSS phenomenon from varied perspectives. For example, focusing on technical perspective, researchers studied issues such as OSS development methodology (e.g., Jørgensen 2001) and coding quality (e.g., Stamelos et el. 2002). Based on social psychology, researchers investigated individual motivation (e.g., Hann et al. 2004), new developers (Von Krogh, Spaeth, & Lakhani 2003), social network (e.g., Madey et al. 2002), and social structure (e.g., Crowston and Howison, 2005). In terms of organizational and managerial perspective, researchers examined knowledge innovation (e.g., Hemetsberger 2004, Lee and Cole 2003, von Hippel and von Krogh 2003) and governance mechanism (e.g., Sagers 2004).

An OSS development team is essentially a virtual organization in which participants interact and collaborate with each other through Internet. Comparing to that of conventional organizations, the structure of virtual organizations is suggested to be decentralized (Raymond, 1998), flat, and non-hierarchical (Ahuja and Carley 1999). However, some researchers challenge the belief (e.g., Crowston and Howison 2005, Gacek 2004, Moon 2000, Mockus 2000, and Mockus 2002). They argue that the social structure of OSS projects is hierarchical rather than flat, like a tree (Gacek 2004) or an onion (Crowston and

Howison 2005). The social structure of OSS teams directly influences the collaboration and the decision-making process, and further affects the overall performance of the teams as well as individual's perception of belongings and satisfaction. Therefore, one wonders what form of social structure might be present in the OSS development, and what type of structure will emerge— centralized or decentralized, hierarchical or non-hierarchical, onion-like or tree-like, or a combination of the above depending on certain specific situations.

A social network, as stated by Krebs and Holley (2004), is generally built in 4 phases, each with its own distinct topology.

1. Scattered Clusters;
2. Single Hub-and-Spoke;
3. Multi-Hub Small-World Network;
4. Core/Periphery.

Most organizations start from isolated and distributed clusters (Krebs and Holley 2004). Then an active leader emerges and takes responsibility to build a network connecting the separate clusters. However, this single hub topology is fragile. With more participants entering the group, the leader changes his/her role to a facilitator and helps to build multiple hubs, which is stage three. The core/periphery model, the last stage, is the most stable structure. In core/periphery structure, network core encompasses key group members that are strongly connected to each other while periphery contains members that are usually weakly connected to each other as well as to the core members. With the general network building phases in mind, one can argue that the OSS projects should follow the same four stages (i.e., scattered clusters, single hub-and-spoke, multi-hub small-world network, core/periphery). But is that true for OSS projects? How does the social structure of OSS projects evolve over time?

Therefore, our research addresses the following questions.

1. What might be the social structure of OSS projects?
2. How the social structure evolves over time?

THEORETICAL FOUNDATION

Social structure and social interaction

Social structure, as suggested by Schaefer and Lamm (1998), refers to the way in which society is organized into predictable relationships. Social structure can be considered in terms of three aspects— actors, their actions, and their interactions. Social actor is a relatively static concept addressing issues such as roles, positions, and status. Individual actors are embedded in the social environment and therefore their actions are largely influenced by the connections between each other. Social interaction is generally regarded as the way in which people respond to one another. These interaction patterns are to some extent independent of individuals. They exert a force that shapes both behavior (i.e., actions) and identity (i.e., actors) (Schaefer and Lamm 1998).

Research on social interaction focuses on how individuals actually communicate with each other in group settings. These studies address issues such as the interaction patterns, the underlying rules guiding interaction, the reasons accounting for the way people interact, and the impacts of the interaction patterns on the individual behavior and the group performance. These issues begin by questioning what might be the interaction pattern in a specific social setting. And that addresses our research question — understanding social interaction of OSS projects.

Social network theory

Social network theory focuses on studying actors as well as their relationships in specific social settings. Network theory is analogous to systems theory and complexity theory. It is an interdisciplinary theory stemming from multiple traditional fields, including psychology which addresses individuals' perception of social structure, anthropology which emphasizes social relationships, and mathematics which provides algorithms (Scott 2000).

Based on the view of social network, the world is composed of actors (also called nodes) and ties between them. The ties can either represent a specific relationship (such as friendship and kinship) between a pair of actors or define a particular action which an actor performs. Different kinds of ties specify different networks and are typically assumed to function differently. For example, the ties in a family network is distinctive from those in a working network, and the centrality in the 'who loves whom" network obviously has different meaning than the centrality in the "who hates whom" network.

Social network theory is based on the intuitive notion that the social interaction patterns are essential to the individuals who reflect them. Network theorists believe that how individuals behave largely depends on how they interact with others and how they are tied to a social network. Furthermore, besides individual behavior, network theorists believe that the success or failure of societies and organizations often depends on the internal interaction pattern (Freeman 2002).

Besides the theoretical essence, social network theory is also characterized as a distinctive methodology encompassing techniques for data collection, statistical analysis, and visual representation. This approach is usually called social network analysis and will be discussed in the research methodology section.

Grounded in the view that actors in OSS development are greatly influenced by the interaction among them, this paper draws on the social network theory to study the interaction pattern of OSS development project.

RESEARCH METHODOLOGY

Social network analysis

Social network analysis is used in our study to investigate the interaction pattern of OSS development process.

Guided by the formal theory organized in mathematical terms and grounded in the systematic analysis of empirical data, social network analysis focuses on uncovering the interaction pattern of interdependent individuals (Freeman 2004). Through a structural analysis of a social network diagram, a map depicting actors as well as ties that connect them together, social network analysis offers promise to reveal the patterns of relationships and the relative position of individuals in a specific social setting. This approach has been effectively used in organizational research, social support, mental health, and the diffusion of information (Freeman 2004). Social network analysis has been used in our study for two primary reasons.

First, the purpose of social network analysis fits our research objective. Social network analysis aims to analyze the relationship among a set of actors instead of their internal attributes. Our research aims to reveal the interaction pattern of OSS projects. Therefore, social network analysis offers great potential to answer our research questions.

Second, the - rich interactive data extracted from OSS projects serves as a reliable and convenient resource for social network analysis. Social network analysis is grounded in the systematic analysis of empirical data. However, usually there is a lack of convenient and objective resource to draw the links (i.e., relationships) among actors. Most OSS projects have online mailing lists, forums, and tracking systems that are open to public, thus providing a rich set of longitudinal data. Based on these public data, researchers are able to generate input data sets for social network analysis.

Longitudinal data

Because we are interested in studying how the interaction pattern of OSS projects evolves over time, single observations of interaction networks are not sufficient. Single observations of social networks are snapshots of the current interaction status and provide untraceable history, thus limiting the usefulness. On the contrary, longitudinal observations offer more promise for understanding the social network structure and its evolution. Therefore, we extracted longitudinal data on OSS projects for this research.

Case selection

OSS projects were selected from the SourceForge[1], which is the world's largest website hosting OSS projects. SourceForge provides free tools and services to facilitate the OSS development. At the time of the study, it hosts a total of 99,730 OSS projects and involves 1,066,589 registered users (data retrieved on May 4th, 2005). Although a few big OSS projects have their own websites (such as Linux), SourceForge serves as the most popular data resource for OSS researchers.

Following the idea of theoretical sampling (Glase & Strauss 1967), three OSS projects were selected from SourceForge in terms of their similarities and their differences. Theoretical sampling requires - theoretical relevance and purposes (Orlikowski 1993). In terms of relevance, the selection ensures the substantive area—the interaction pattern of OSS projects over time—is kept similar. Therefore, the projects that are selected have to satisfy two requirements. First, the projects have considerable interaction among members during the development process. All three projects have larger than 10 developers and the number of bug reported is more than 1000. Second, since we are interested in the interaction over time, the projects should have a relatively long life. In our case, all three projects are at least three years old.

---

[1] The web address for SourceForge is www.sourceforge.net.

In addition to similarities, differences are sought among cases because the study aims to explore interaction pattern applicable to various OSS projects. Therefore, the three projects differ on several project characteristics, such as project size, project type, and intended audience. These differences enable us to make useful contrasts during data analysis.

The following table summarizes the three projects with a brief description.

		Net-SNMP	Compiere ERP + CRM	J-boss
Description		Net-SNMP allows system and network managers to monitor and manage hosts and network devices.	Compiere is a smart ERP+CRM solution covering all major business areas - especially for small-medium Enterprises.	JBoss is a leading open source java application server. After Linux and Apache, it is the third major open source project to receive widespread adoption by corporate IT.
Similarities	Bug reports (over 1000 bugs)	1361	1695	2296
	Development duration (over 3 years)	55 month (registered on 10/2000)	47 month (registered on 6/2001)	50 month (registered on 3/2001)
Differences	Software type	Internet, network management	Enterprise: ERP+CRM	J2EE-based Middleware
	Group size (the number of developers)	Small (14)	Median (44)	Large (75)
	Intended audience	Developers, system administrators	Business	Developers, system administrators

Table 1. Summary of three projects (data achieved in April 2004 from SourceForge.net)

Data collection and analysis

Social network analysis can be divided into the following three stages (Borgatti 2002).

1. Data collection. In this stage, researchers collect data using surveys and questionnaires or from documents and other data resources, and generate input data sets for social network analysis.

2. Statistical analysis. Based on mathematics algorithms, this stage generates network indices concerning group structure (such as centralization and density) as well as individual cohesion (such as centrality and bridges).

3. Visual representation. This stage employs network diagram to show the interaction structure as well as the position of specific actors.

First is the data collection. The data were collected in April 2005 from SourceForge.net. Data were extracted from the bug tracking system of each project. We chose the bug tracking system as the primary data resource for three reasons. First, open source software is characterized as peer review of open codes. Raymond (1998) proposed the “Linus’ law” in his well known essay “The Cathedral and the Bazaar”, “Given enough eyeballs, all bugs are shallow.” Therefore, the bug system can be viewed as the representative of open source spirit. Second, comparing to other development activities such as patch posting and feature request, the bug-fixing process is the most active procedure to illustrate the close collaboration between developers and users as well as between developers themselves. Finally, the bug tracking system provides rich data that record the interactive procedure.

A web spider program, which is based on the work of Crowston and Howison (2005) with necessary revision, is used to download the bug tracking data from the project website. The interaction data is extracted from the bug tracking system month-by-month starting from the date the project is registered to the date the data is downloaded for this study. The output of this stage is a matrix describing the interaction among users. Each row or column of the matrix represents a distinctive participant, which is identified by a unique SourceForge user ID. The value in the matrix indicates the interaction between each pair of participants, which is counted by the amounts of messages that participant A (i.e., row A) replied to participant B (i.e., column B).

Second is the statistical analysis. Our study focuses on two important and distinctive properties of network structure—group centralization and core/periphery fitness. Ucinet, which was developed by Borgatti, Everett, and Freeman (2002), has been used to calculate these two properties.

Group centralization, as suggested by Wasserman & Faust (1994), refers to the extent to which a network revolves around a single center. A typical case of centralized structure is a "star" network. Group centralization can be viewed as a rough measure of inequity between individual actors, and the variability and dispersion of the interaction pattern.

The other property is core/periphery fitness. It measures the extent to which the network is close to a perfect core/periphery structure. The core/periphery structure depicts a dense, connected group surrounded by a sparse, unconnected periphery. The opposite structure is clique, which represents a structure of multiple sub groups and each with its own core and peripheries (Borgatti 2002).

Finally is the visual representation. We used Ucinet (Borgatti et al. 2002) to draw the interaction networks for each of the three projects.

RESEARCH RESULTS

Snapshots of the three projects

Monthly data were extracted from the bug tracking system of each project. To provide a rough image of the trend of interaction pattern, we provide three snapshots for each project[2].

Table 2 summarizes the relevant network characteristics of each project. In addition to the group centralization and core/periphery fitness, we also report other network characteristics such as density, average distance, and distance-based cohesion. Density depicts how "close" the network looks and it is a recommended measure of group cohesion (Blau 1977, Wasserman & Faust 1994). The value of density ranges from 0 to 1. Average distance refers to average distance between all pairs of nodes (Borgatti 2002). Distance-based cohesion takes on values between 0 and 1; the larger the values, the greater the cohesiveness.

		Net-SNMP	Compiere	JBoss
Group centralization (%)	1st.	9.420	15.624	4.931
	2nd.	3.071	2.294	4.45
	3rd.	2.316	1.288	4.12
Core/Periphery Fitness	1st.	0.674	0.774	0.485
	2nd.	0.654	0.796	0.477
	3rd.	0.651	0.765	0.501
Density	1st.	0.0235	0.0584	0.0073
	2nd.	0.0109	0.0610	0.0039
	3rd.	0.0072	0.0571	0.0026
Average distance	1st.	2.546	2.711	3.438
	2nd.	2.794	2.302	3.281
	3rd.	2.917	2.278	3.239
Distance-based cohesion	1st.	0.181	0.198	0.118
	2nd.	0.143	0.253	0.147
	3rd.	0.141	0.279	0.136

Table 2. Three snapshots for each project

Figures 1, 2 and 3 provide visual representation of the interaction pattern on the three time stamps for each project.

[2] Time was divided equally into three slides for each project. The three time stamps for Net-SNMP are 4/2002, 10/2003, 4/2005, for Compiere are 9/2002, 1/2004, 4/2005, and for Jboss are 7/2002, 11/2003, 4/2005.

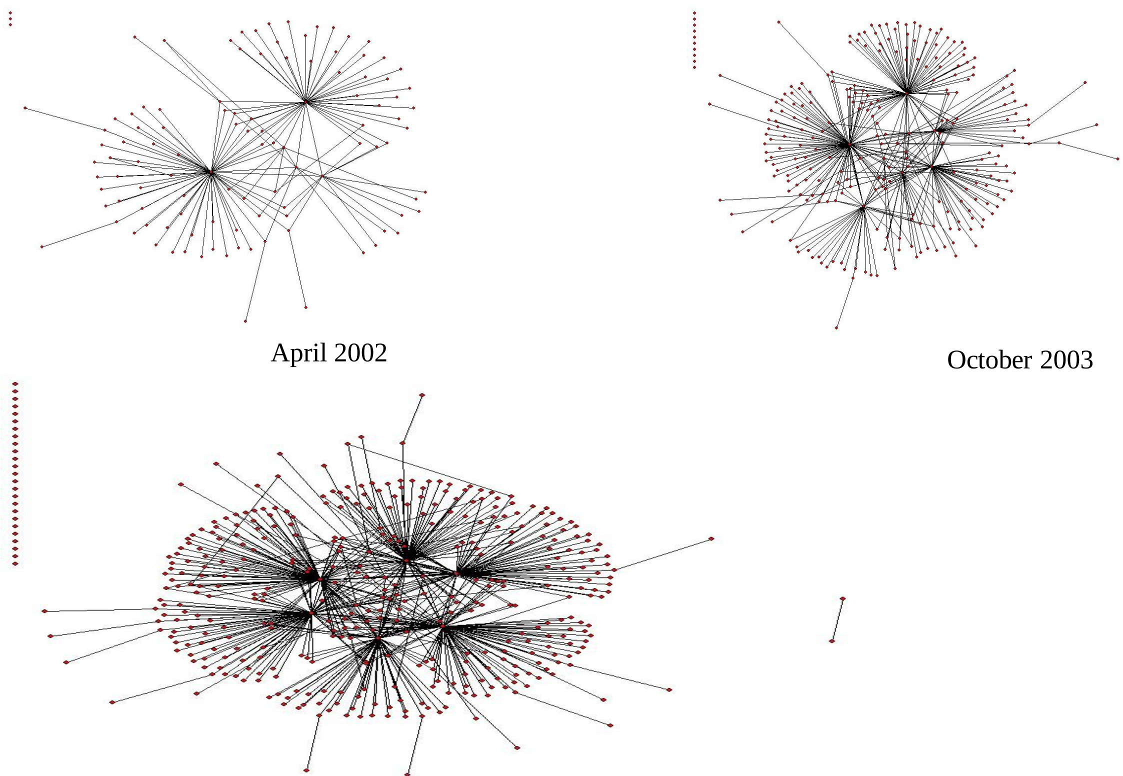

Figure 1. Plots of the interaction for Net-SNMP

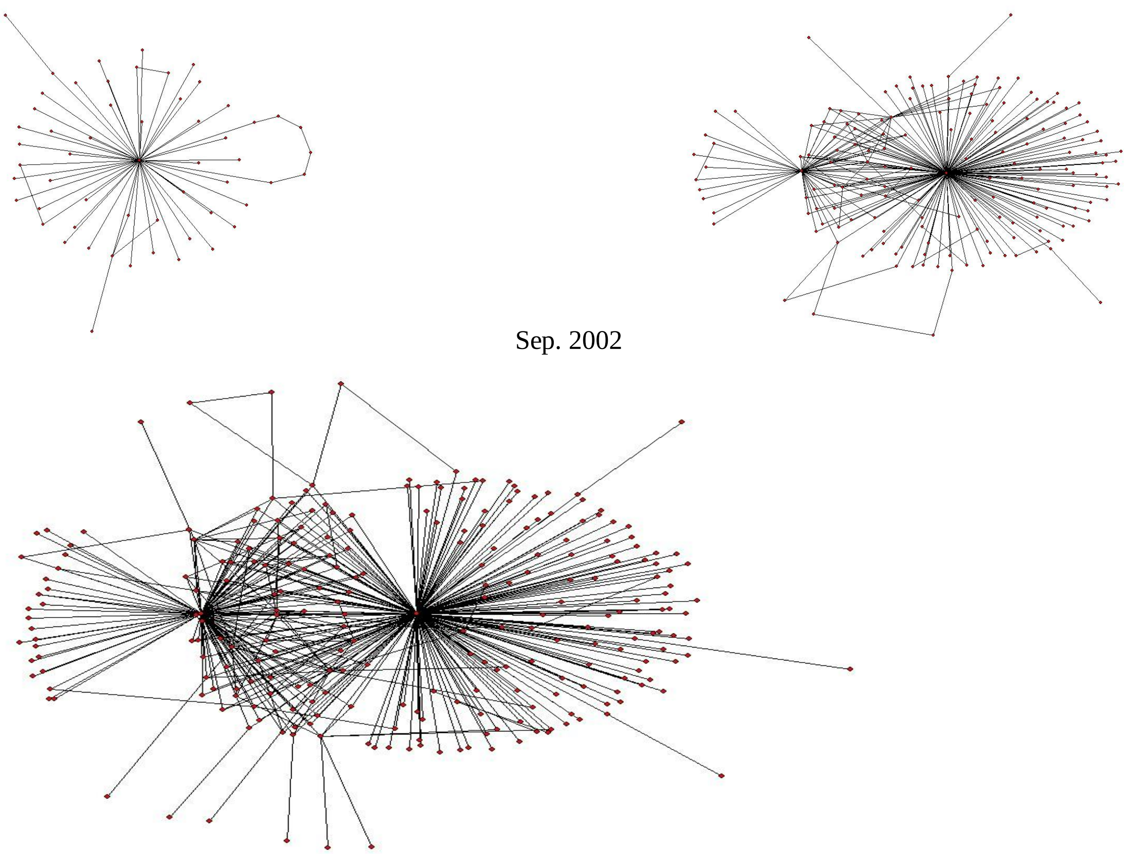

Figure 2. Plots of the interaction for Compiere CRM+ERP

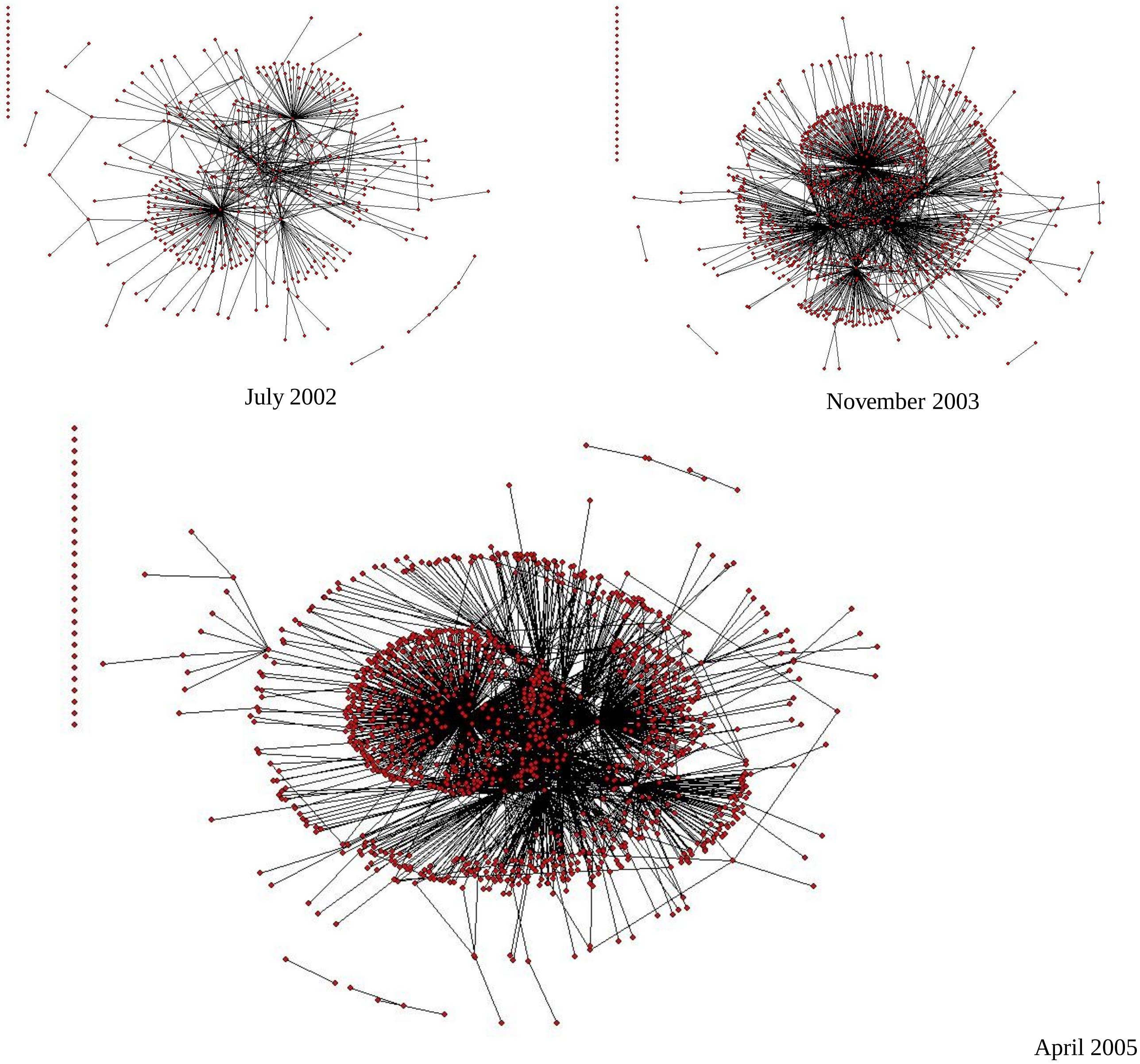

Figure 3. Plots of the interaction for JBoss

Looking at the statistical analysis and the network plots, we can observe the following:

First, the evolvement of interaction pattern of the three projects reveals a general trend. As shown in the network plots (i.e., figures 1, 2, and 3), the interaction pattern develops from a centralized one— single (sometimes two) hub with several distributed nodes, to a core/periphery structure— a core (a group of core developers) together with several hangers-on (periphery). Condense interactions exist within the core (among several core developers) and between each core member and his/her periphery. However, only loose relationships exist among peripheries. This pattern suggests a layer structure (i.e., core with its periphery) instead of a complete flat one with equal positions across all the members.

Second, although the interaction patterns of the three projects share some commonalities, the shape of the structures is slightly different. The shape of Net-SNMP (as shown in figure 1) is more like a typical core/periphery compared to the other two. Compiere (as shown in figure 2) keeps two cores, and the shape looks like a dumbbell. In addition, Jboss (as shown in figure 3), which is the largest project among the three, maintains a more complex structure which shows multiple layers instead of just one core with the rest as peripheries (e.g., Net-SNMP)

Third, as time goes by, the group centralization decreases across the three projects, showing a trend from a centralized structure to a decentralized structure irrespective of project size (the three projects are with different project sizes as shown in table 1), project types, and intended audience.

Fourth, the indices of core/periphery fitness of each project fluctuate slightly but maintain a relatively high value (larger than 0.5 on average). However, no observable trend was found across projects.

Fifth, since each of the projects has a relatively large group (i.e., over 100 actors including all the registered users), the values of density are relatively low with little variation. Therefore, density is not appropriate to be used to compare the projects.

From the snapshots, we observed the following trend. First, OSS interaction network develops into a core/periphery structure; second, group centralization decreases over time. Third, core/periphery fitness stays relatively stable. To verify our observations, we used longitudinal data generated from the bug tracking systems to analyze the evolvement of interaction pattern (discussed in the following section).

Group centralization and Core/periphery fitness

Figure 4 shows the values of both group centralization and core/periphery fitness along time based on the monthly interaction data. For each figure, the Y-axis indicates the indices (i.e., group centralization or core/periphery fitness), and the X-axis reflects the time dimension starting from the date the project is registered on SourceForge.

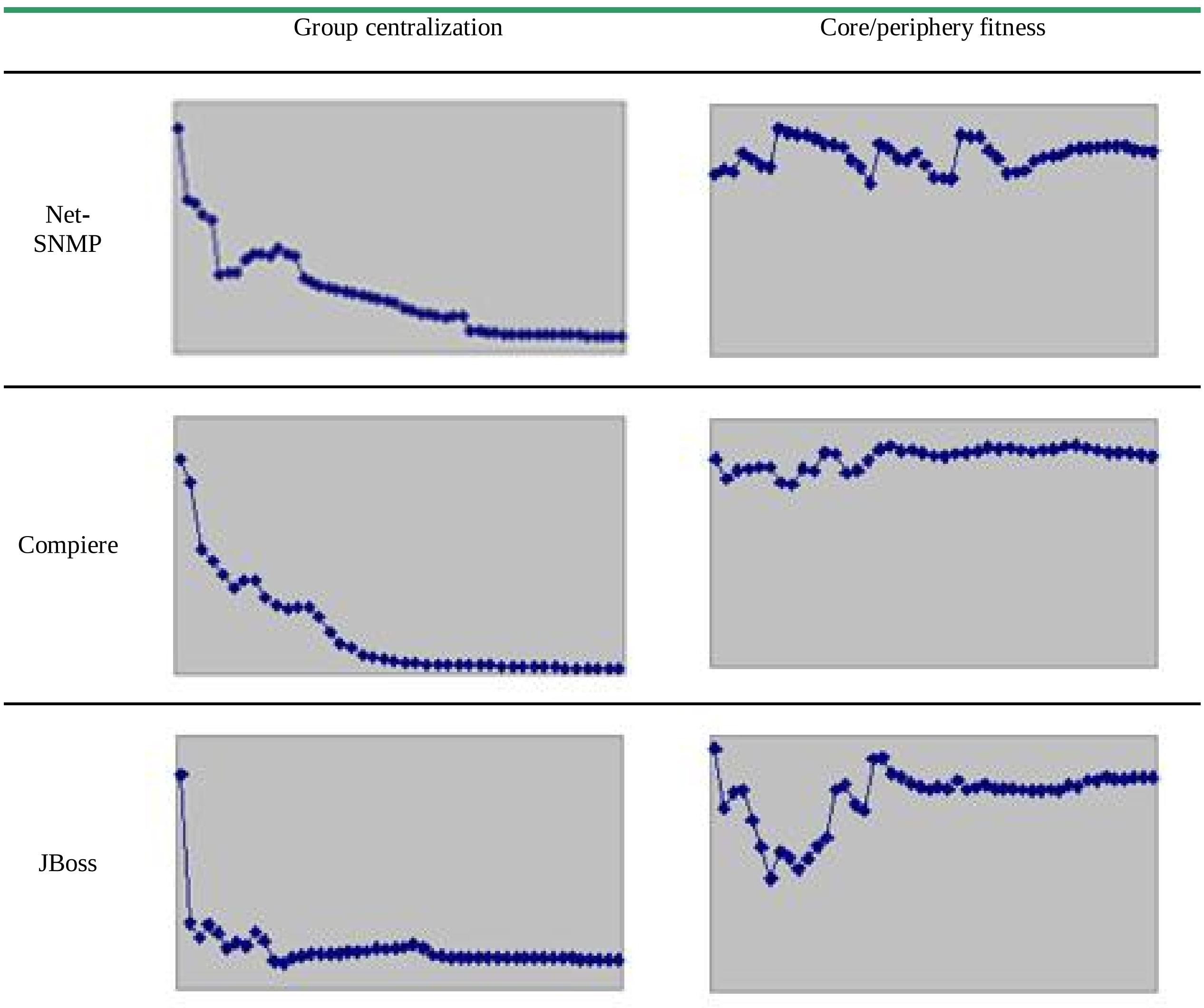

Figure 4. The trend of group centralization and core/periphery fitness based on longitudinal data

Two primary observations can be made based on the statistical analysis:

First, the group centralization shows a decreasing trend across the three projects. This observation indicates that as OSS projects progress, the social network structure evolves from centralized to decentralized, and then stabilizes. Also, the three figures suggest no substantial differences in the trend between the three projects.

Second, the core/periphery index is maintained at a relatively stable level for each project over time. In addition, the average fitness value stays relatively high for each project (larger than 0.5), indicating a closeness to a perfect core/periphery structure.

Besides a holistic view of network structure for OSS projects, the results also reveal other interesting findings. For example, by examining the core members over time, we found a relatively stable core for each project. The cores are usually project developers and administrators. This observation further demonstrates the existence of strong and stable core as well as loose hangers-on in OSS projects.

DISCUSSIONS

Using social network analysis, this research studies the longitudinal data of three OSS projects selected from SourceForge. The purpose of this study is to investigate the evolvement of interaction pattern of OSS projects. The research results suggest a decreasing of group centralization over time and a tendency of core/periphery structure in OSS projects.

The network plots (as shown in figures 1, 2 and 3) indicate a layer structure instead of a flat one suggested by earlier literature. The interaction pattern evolves from a single hub to a core/periphery structure. As the number of participants increases, a core with only one person (may be the starter/initiator of the project) cannot satisfy the increasing requirements of development and communication. Therefore, other developers or active users join the core to serve as key members of the project. This result in a more stable structure and the project is not be affected by a single leader.

The specific research setting we chose may also be part of the reason leading to the core/periphery model. Bug tracking system was selected as our research setting and data resource. With the growth of a software project, more people are attracted to the project. The original leader may not be able to solve all the technical problems encountered in the development process. Each key member has his/her specialty and is responsible for solving relevant problems, and has his/her own periphery in the network plot. Although there are multiple peripheries in the project, collaboration among key members in the project is vital. This phenomenon of distribution and collaboration can be viewed as a critical success factor of OSS development. And the evolvement is vividly demonstrated in our social network analysis.

It is inappropriate to make general claims on whether the social structure of OSS projects is centralized or decentralized. On one hand, it is centralized in the sense that there exists a core which consists of key members. These key members are responsible for various issues encountered during the development process. On the other hand, it is decentralized in the sense that the decision or communication core is not concentrated on one or two members but a group of key members.

Like any other research, this research also has a few limitations. First, the cases were only selected from SourceForge.net. Although SourceForge is the world's largest website hosting open source software, there also exist some other similar websites. Therefore, the total number of OSS projects on Sourcefore cannot be viewed as the whole population. However, as we argued before, SourceForge is probably the best data collection site for this research.

Second, the bug tracking system was chosen as our data resource. Besides the bug tracking forum, there are other forums that also provide space for participants to communicate with one another, such as mailing list and feature request. However, as we argued earlier, the bug systems are the most active forum providing rich interaction data. The bug tracking systems also represents the spirit of open source software. Examining the interaction data of other forums can be one of our research extensions in the future.

Third, because our research objective is to investigate interaction pattern, we chose projects that have a relatively large number of developers, a large amount of bug reports, and relatively long history. Although we tried to involve different types of projects (i.e., different project sizes, project types, and intended audience), these three cases may not be representative of all OSS projects — especially very smaller projects with limited interactions. Increasing the sample size and including more types of OSS projects is one of our future research directions.

IMPLICATIONS AND CONCLUSIONS

This paper examines the evolvement of interaction pattern of OSS projects. Our research findings suggest that the interaction structure starts from a single hub and evolves to a core/periphery model. We argue that the social structure of OSS project is both centralized and decentralized. It is centralized in the sense that there exists a relatively stable core consisting of key developers. It is also decentralized because of distributed decision-making among key developers, and the broad collaboration between developers and users as well as among developers themselves.

The paper presents the evolvement of the social structure of OSS projects from a longitudinal perspective. It also provides empirical evidence of the change of interaction pattern from a single hub to a core/periphery model. Moreover, the paper utilizes social network analysis as the research method. This approach has been shown in this research as an effective tool to analyze social structure in OSS projects.

Social structure is an important variable to understand social phenomenon. Open source software, with its open and unique nature, attracts researchers to ask a series of why and how questions. For example, how participants of OSS projects collaborate and interact with each other? How they control and manage their development process? Why are some projects more successful than others? Social network analysis is a good approach to answer these questions and study the dynamics of OSS development. This research represents a pioneering effort in this direction

REFERENCE

1. Ahuja, M. and Carley, K. (1999). Network Structure in Virtual Organizations, *Organization Science*, 10(6), 741-747
2. Blau, P.M. (1977). *Inequity and Heterogeneity*. New York: Free Press.
3. Borgatti, S. (2002). *Basic social network concepts*. http://www.analytictech.com/networks/basic%20concepts%202002.pdf
4. Borgatti, S.P., Everett, M.G. and Freeman, L.C. (2002). Ucinet for Windows: Software for Social Network Analysis. Harvard, MA: Analytic Technologies.
5. Crowston, K. and Howison, J. (2005) The social structure of Free and Open Source software development, *First Monday*, 10(2)
6. Freeman, L.C. (2004). *The Development of Social Network Analysis: A Study in the Sociology of Science*. Vancouver: Empirical Press.
7. Gacek, C., & Arief, B. (2004). The many meanings of open source. *IEEE Software*, *21*(1), 34–40.
8. Hann, H., Robert J. and Slaughter, S. (2004) Why Developers Participate in Open Source Software Projects: An Empirical Investigation. In Twenty-Fifth *International Conference on In- formation Systems*, Washington, DC, pp. 821-830,
9. Hemetsberger, A. (2004) Sharing and Creating Knowledge in Open-Source Communities: The case of KDE The Fifth European Conference on Organizational Knowledge, Learning, and Capabilities in Innsbruck, Austria.
10. Jørgensen, N. (2001) Putting it all in a trunk: incremental software development in the FreeBSD open source project. *Information Systems Journal*. 11 321-336.
11. Krebs, V. and Holley, J. (2004). Building Sustainable Communities through Network Building, http://www.orgnet.com/BuildingNetworks.pdf
12. Lee, G.K. and Cole, R.E. (2003). “From a Firm-Based to a Community-Based Model of Knowledge Creation: The case of the Linux Kernel Development”. Organization Science, 14, 6, pp.633-649.
13. Madey, G., Freeh, V., Tynan R. (2002). The Open Source Software Development Phenomenon: An Analysis Based on Social Network Theory, AMCIS2002, Dallas, TX.
14. Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2000). A case study of open source software development: The apache server. ICSE 2000.
15. Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and mozilla. ACM Transactions on Software Engineering and Methodology, 11(3), 309–346.
16. Moon, J. Y., & Sproull, L. (2000). Essence of distributed work: The case of linux kernel. First Monday, 5(11).

17. Orlikowski, W. J. (1993). CASE tools as organizational change: investigating incremental and radical changes in systems development. MIS Quarterly, 17(3), 309-340.
18. Raymond, E.S. (2000). *The Cathedral and the Bazaar*. http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/. Access in Jan. 2005
19. Sagers, G.W. (2004) The influence of network governance factors on success in open source software development projects. In Twenty-Fifth *International Conference on In- formation Systems*, Washington, DC, pp. 427–438
20. Schaefer, R. T. and Lamm, R.P. (1998). *Sociology*. 6th edition. McGraw-Hill
21. Scott, J. (2000). *Social Network Analysis. A Handbook.* SAGE Publications, London, U.K., second edition
22. Stamelos, I., Angelis, L., Oikonomu, A., and Bleris, G.L. (2002) Code quality analysis in Open-Source software development. *Information Systems Journal*, 12(1): 43-60
23. Von Hippel, E., G. Von Krogh. (2003). Open source software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science*. 14 209-223.
24. Von Krogh, G., Spaeth, S., Lakhani, K. R. 2003. Community, joining, and specialization in open source software innovation: A case study. Research Policy, 32(7): 1217-1241.