Drawing the Big Picture: Temporal Visualization of Dynamic Collaboration Graphs of OSS Software Forks

Amir Azarbakht and Carlos Jensen

Oregon State University, School of Electrical Engineering & Computer Science,
1148 Kelley Engineering Center, Corvallis OR 97331, USA
{azarbaam,cjensen}@eecs.oregonstate.edu

Abstract. How can we understand FOSS collaboration better? Can social issues that emerge be identified and addressed as they happen? Can the community heal itself, become more transparent and inclusive, and promote diversity? We propose a technique to address these issues by quantitative analysis and temporal visualization of social dynamics in FOSS communities. We used social network analysis metrics to identify growth patterns and unhealthy dynamics; This gives the community a heads-up when they can still take action to ensure the sustainability of the project.

1 Introduction

Social networks are a ubiquitous part of our social lives, and the creation of online social communities has been a natural extension of this phenomena. Free/Open Source Software (FOSS) development efforts are prime examples of how community can be leveraged in software development, groups are formed around communities of interest, and depend on continued interest and involvement in order to stay alive [17].

Though the bulk of collaboration and communication in FOSS communities occurs online and is publicly accessible, there are many open questions about the social dynamics in FOSS communities. Projects might go through a metamorphosis when faced with an influx of new developers or the involvement of an outside organization. Conflicts between developers' divergent visions about the future of the project might lead to forking of the project and dilution of the community. Forking, either as a violent split when there is a conflict or as a friendly divide when new features are experimentally added both affect the community [3].

Most recent studies of FOSS communities have tended to suffer from an important limitation. They treat community as a static structure rather than a dynamic process. In this paper, we propose to use temporal social network analysis to study the evolution and social dynamics of FOSS communities. With these techniques we aim to identify measures associated with unhealthy group dynamics, e.g. a simmering conflict, as well as early indicators of major events

L. Corral et al. (Eds.): OSS 2014, IFIP AICT 427, pp. 41–50, 2014.
© IFIP International Federation for Information Processing 2014

in the lifespan of a community. One set of dynamics we are especially interested in, are those that lead FOSS projects to fork. We used the results of a study of forked FOSS projects by Robles and Gonzalez-Barahona [19] as the starting point for out study, and tried to gain a better understanding of the evolution of these communities.

This paper is organized as follows: We present related literature on online social communities. We then present the gap in the literature, and discuss why the issue needs to be addressed. After that, in methodology, we describe how gathering data, doing the analysis, and the visualization of the findings was carried out. At the end, we present results, discussion and threats to validity.

2 Related Work

The social structures of FOSS communities have been studied extensively. Researchers have studied the social structure and dynamics of team communications [4][10][11], identifying knowledge brokers and associated activities [20], project sustainability [10], forking [18] [19], their topology [4], their demographic diversity [13], gender differences in the process of joining them [12] and the role of the core team in their communities [21], etc. All of these studies have tended to look at community as a static structure rather than a dynamic process. This makes it hard to determine cause and effect, or the exact impact of social changes.

The study of communities has grown in popularity in part thanks to advances in social network analysis. From the earliest works by Zachary [22] to the more recent works of Leskovec et al. [14][15], there is a growing body of quantitative research on online communities. The earliest works on communities was done with a focus on information diffusion in a community [22]. Zachary investigated the fission of a community, the process of communities splitting into two or more parts. He found that fission could be predicted by applying the Ford-Fulkerson min-cut algorithm [6] on the group's communication graph; "the unequal flow of sentiments across the ties" and discriminatory sharing of information lead to "subcommunities with more internal stability than the community as a whole."

Community splits in FOSS are referred to as forks, and are relatively common. Forking is defined as "when a part of a development community (or a third party not related to the project) starts a completely independent line of development based on the source code basis of the project." Robles and Gonzalez-Barahona [19] identified 220 significant FOSS projects that have forked over the past 30 years, and compiled a comprehensive list of the dates and reasons for forking. They classified these into six main categories. (Table 3.) which we build on extensively. They identified a gap in the literature in case of "how the community moves when a fork occurs".

The dynamic behavior of a network and identifying key events was the aim of a study by Asur et al [1]. They studied three DBLP co-authorship networks and defined the evolution of these networks as following one of these paths: a) Continue, b) k-Merge, c) k-Split, d) Form, or e) Dissolve. They also defined four possible transformation events for individual members: 1) Appear, 2) Disappear,

3) Join, and 4) Leave. They compared groups extracted from consecutive snapshots, based on the size and overlap of every pair of groups. Then, they labeled groups with events, and used these identified events. The communication patterns of FOSS developers in a bug repository were examined by Howison et al. [10]. They calculated out-degree centrality as their metric. Out-degree centrality measures the proportion of the number of times a node contacted other nodes (outgoing) over how many times it was contacted by other nodes (incoming). They calculated this centrality over time "in 90-day windows, moving the window forward 30 days at a time." They found that "while change at the center of FOSS projects is relatively uncommon," participation across the community is highly skewed, following a power-law distribution, where many participants appear for a short period of time, and a very small number of participants are at the center for long periods. Our approach is similar to theirs in how we form collaboration graphs and perform our temporal analysis. Our approach is different in terms of our project selection criteria, the metrics we examine, and our research questions.

The tension between diversity and homogeneity in a community was studied by Kunegis et al. [13]. They defined five network statistics used to examine the evolution of large-scale networks over time. They found that except for the diameter, all other measures of diversity shrunk as the networks matured over their lifespan. Kunegis et al. [13] argued that one possible reason could be that the community structure consolidates as projects mature.

Community dynamics was the focus of a recent study by Hannemann and Klamma [8] on three open source bioinformatics communities. They measured "age" of users, as starting from their first activity and found survival rates and two indicators for significant changes in the core of the community. They identified a survival rate pattern of 20-40-90%, meaning that only 20% of the newcomers survived after their first year, 40% of the survivors survived through the second year, and 90% of the remaining ones, survived over the next years. As for the change in the core, they suggested that a falling maximal betweenness in combination with an increasing network diameter as an indicator for a significant change in the core, e.g. retirement of a central person in the community. Our approach builds on top of their findings, and the evolution of betweenness centralities and network diameters for the projects in our study are depicted in the following sections.

To date, most studies on FOSS have only been carried out on a small number of projects, and using snapshots in time. To our knowledge, no study has been done of project forking that has taken into account the temporal dimension.

3 Methodology

We argue that the social interactions data reflects the changes the community goes through, and will be able to describe the context surrounding a forking event. Robles and Gonzalez-Barahona [19] classify forking into six classes, listed in Table 1, based on the motives for forking.

Table 1. The main reasons for forking as classified by Robles and Gonzalez-Barahona [19]

Reason for forking	Example forks
Technical (Addition of functionality)	Amarok & Clementine Player
More community-driven development	Asterisk & Callweaver
Differences among developer team	Kamailio & OpenSIPS
Discontinuation of the original project	Apache web server
Commercial strategy forks	LibreOffice & OpenOffice.org
Legal issues	X.Org & XFree

The first three of the six motives listed are social, and so should arguably be reflected in the social interaction data. For example, if a fork occurs because of a desire for "more community-driven development", we would perhaps expect to see patterns in the data showing a strongly-connected core that is hard to penetrate for the rest of the community prior to the fork. In other words, the power stays in the hands of the same people over a long period of time while new people come and go. Our goal was to visualize and quantify how the community is structured, how it evolves, and the degree to which involvement changes over time. To this end, we picked projects from the aforementioned three categories of forked projects. This involved obtaining communication archives, creating the collaboration graphs, applying social network analysis (SNA) techniques to measure key metrics, and visualizing the evolving graphs. We did this in four phases as described in the following:

3.1 Phase 1: Data Collection

The study of forks by Robles and Gonzalez-Barahona [19] included information on 220 forks and their reasons. We applied three selection criteria to those projects. A project was short-listed if it was recent, i.e. the fork had happened after the year 2000; data was available; and their communities were of approximately the same size. This three stage filtering process resulted in the projects listed in Table 2.

Data collection involved analyzing mailing list archives. We collected data for the year in which the fork happened, as well as for three month before and three

Table 2. Forked projects for which collaboration data was collected

Projects	Reason for forking	Year
Amarok & Clementine Player	Technical (Addition of functionality)	2010
Asterisk & Callweaver	More community-driven development	2007
Kamailio & OpenSIPS	Differences among developer team	2008

months after that year in order to capture the social context context at the time of the fork.

3.2 Phase 2: Creating Communication Graphs

Many social structures can be represented as graphs. The nodes represent actors/players and the edges represent the interaction between them. Such graphs can be a snapshot of a network – a static graph – or a changing network, also called a dynamic graph. In this phase, we processed the data to form a communication graph of the community. We were looking for how people interacted with each other. We decided to treat the general mailing list as a person, because the bulk of the communication was targeted at it, and most newcomers start by sending their questions to the general mailing list. Each communication effort was captured with a time-stamp. This allowed us to form a dynamic graph, in which the nodes would exist if and only if they had an interaction with another node during the period we were interested in.

3.3 Phase 3: Temporal Visualization and Temporal Evolution Analysis

In this phase, we wanted to analyze the changes that happen to the community over a given period of time, i.e. three months before and three months after the year in which the forking event happened. We measured betweenness centrality [5] of the most significant nodes in the graph, and the graph diameter over time. Figures 2-4 show the betweenness centralities over the 1.5 year period for the Camailio, Amarok and Asterisk projects respectively. To do temporal analysis, we had two options; 1) look at snapshots of the network state over time, (e.g. to look at the network snapshots in every week, the same way that a video is composed of many consecutive frames), and 2) look at a period through a time window. We preferred the second approach, and looked through a time window of three months wide with 1.5 month overlaps. To create the visualizations, we used a 3 months time frame that progressed six days a frame. In this way, we had a relatively smooth transition.

We visualized the dynamic network changes using Gephi [2]. The videos show how the community graph is structured, using a continuous force-directed linear-linear model, in which the nodes are positioned near or far from each other proportional to the graph distance between them. This results in a graph shape between between Früchterman & Rheingold's [7] layout and Noack's LinLog [16].

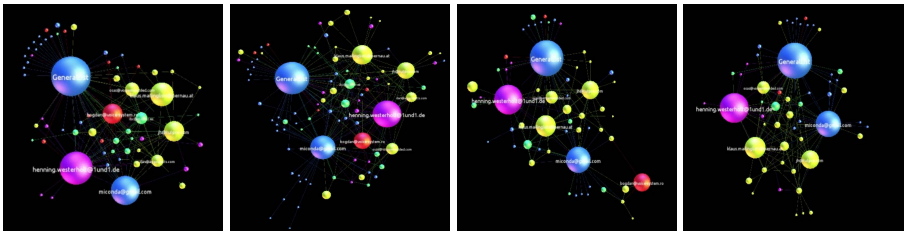
4 Results and Discussion

4.1 Kamailio Project

Figure 1 shows four key frames from the Kamailio project's social graph around the time of their fork (the events described here are easier to fully grasp by

watching the video. A node's size in a proportional to the number of interactions the node (contributor) has had within the study period and the position and edges of the nodes change if they had interactions within the time window shown, with six day steps per frame. The 1 minute and 37 seconds video shows the life of the Kamailio project between October 2007, and March 2009. Nodes are colored based on the modularity of the network.

The community starts with the GeneralList as the the biggest node, and four larger core contributors and three lesser size core contributors. The big red-colored node's transitions are hard to miss, as this major contributor departs from the core to the periphery of the network (Video minute 1:02) and then leaves the community (Video minute 1:24) capturing either a conflict or retirement. This corresponds to the personal difference category of forking reasons.



(a) (b) (c) (d)

Fig. 1. Snapshots from video visualization of Kamailio's graph (Oct. 2007 - Mar. 2009) in which a core contributor (colored red) moves to the periphery and eventually departs the community

Figure 2 shows the betweenness centrality of the major contributors of Kamailio project over the same time period. The horizontal axis marks the dates, (each mark represents a 3-month time window with 1.5 months overlap). The vertical axis shows the percentage of the top betweenness centralities for each node. The saliency of the GeneralList – colored as light blue – is apparent due to to its continuous and dominant presence in the stacked area chart. The chart legend lists the contributors based on the color and in the same order of appearance on the chart starting from the bottom. One can easily see that around the "Aug. 15, 2008 - Nov. 15, 2008" tick mark on the horizontal axis, several contributors' betweenness centralities shrink to almost zero and disappear. This helps identify the date of fork with a month accuracy. The network diameter of the Kamailio project over the same time period is also shown in Figure 3. The increase in the network diameter during this period confirms the findings of Hannemann and Klamma [8].

This technique can be used to identify the people involved in conflict and the date the fork happened with a months accuracy, even if the rival project does not emerge immediately.

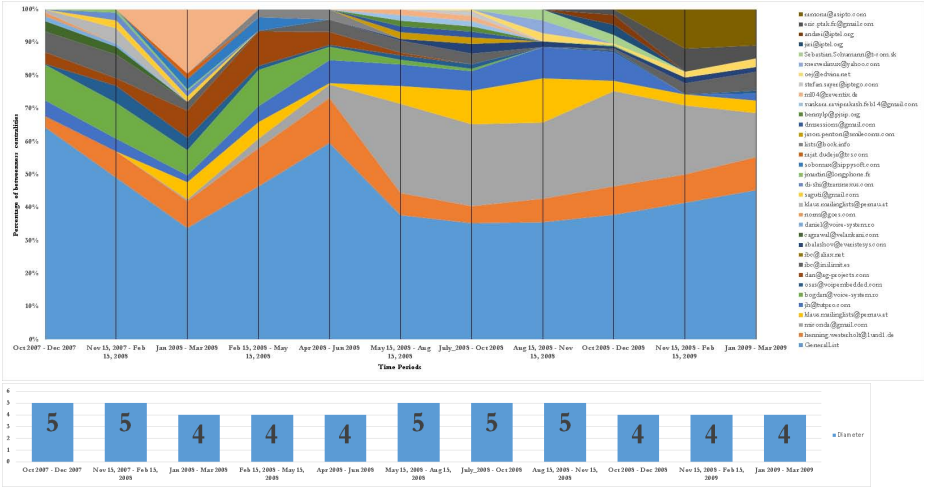


Fig. 2. Kamailio top contributors' betweenness centralities and network diameter over time (Oct. 2007 to Mar. 2009) in 3-month time windows with 1.5-month overlaps

4.2 Amarok Project

The video for the Amarok project fork is available online[1], and the results from our quantitative analysis of the betweenness centralities and the network diameters are shown in Figure 3. The results show that the network diameter has not increased over the period of the fork, which shows a resilient network. The video shows the dynamic changes in the network structure, again typical of a healthy network, rather than of simmering conflict. These indicators show that Amarok fork in 2010 arguably belongs to the "addition of technical functionality" rationale for forking, as there are no visible social conflict.

4.3 Asterisk Project

The video for the Asterisk project is also available online, and the results from our quantitative analysis of the betweenness centralities and the network diameters are shown in Figure 4. The results show that the network diameter remained steady at 6 throughout the period. The Asterisk community was by far the most crowded project, with 932 nodes and 4282 edges. The stacked area chart shows the distribution of centralities, where we see an 80%-20% distribution (, i.e. 80% or more of the activity is attributed to six major players, with the rest of the community accounting for only 20%). This is evident in the video representation as well, as the top-level structure of the network holds throughout the time period. The results from the visual and quantitative analysis links the Asterisk fork to the more community-driven category of forking reasons.

[1] Video visualizations available at
`http://eecs.oregonstate.edu/~azarbaam/OSS2014/`

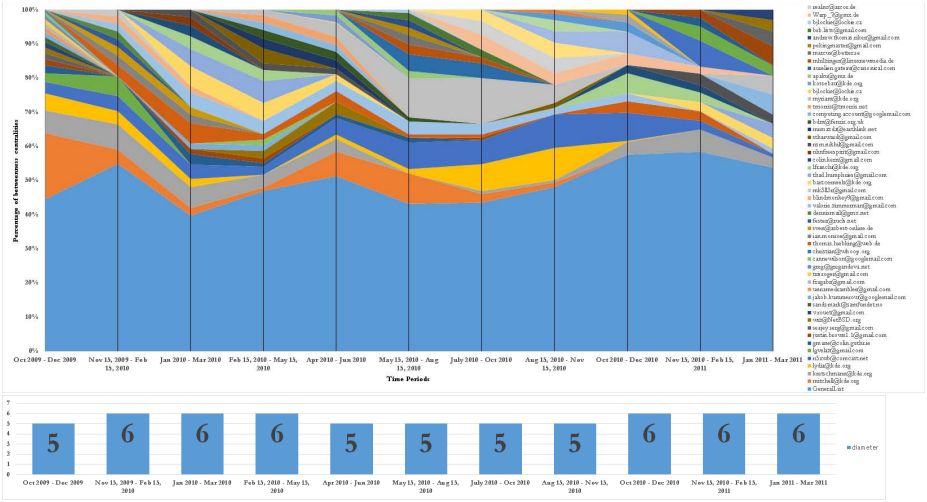


Fig. 3. Amarok project's top contributors' betweenness centralities and network diameter over time between Oct. 2009 to Mar. 2011 in 3-months time windows with 1.5 months overlaps

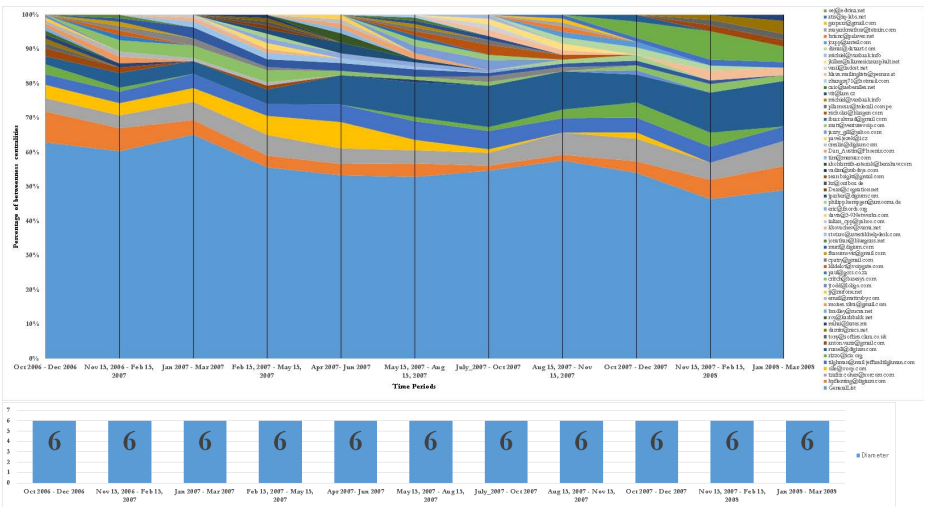


Fig. 4. Asterisk project's top contributors' betweenness centralities and network diameter over time between Oct. 2009 to Mar. 2011 in 3-months time windows with 1.5 months overlaps

5 Conclusion

We studied the collaboration networks of three FOSS projects using a combination of temporal visualization and quantitative analysis. We based our study on two papers by Robles and Gonzalez-Barahona [19] and Hannemann and Klamma [8], and identified three projects that had forked in the recent past. We mined the collaboration data, formed dynamic collaboration graphs, and measured social network analysis metrics over an 18-month period time window.

We also visualized the dynamic graph (available online) and as stacked area charts over time. The visualizations and the quantitative results showed the differences among the projects in the three forking reasons of personal differences among the developer teams, technical differences (addition of new functionality) and more community-driven development. The personal differences representative project was identifiable, and so was the date it forked, with a month accuracy. The novelty of the approach was in applying the temporal analysis rather than static analysis, and in the temporal visualization of community structure. We showed that this approach shed light on the structure of these projects and reveal information that cannot be seen otherwise.

References

1. Asur, S., Parthasarathy, S., Ucar, D.: An event-based framework for characterizing the evolutionary behavior of interaction graphs. ACM Trans. Knowledge Discovery Data 3(4), Article 16, 36 p. (2009)
2. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks. Presented at the Int. AAAI Conf. on Weblogs and Social Media (2009)
3. Bezrukova, K, Spell, C.S., Perry, J.L.: Violent Splits Or Healthy Divides? Coping With Injustice Through Faultlines. Personnel Psychology 63(3) (2010)
4. Bird, C., Pattison, D., D'Souza, R., Filkov, V., Devanbu, P.: Latent social structure in open source projects. In: Proc. of the 16th ACM SIGSOFT Int. Symposium on Foundations of Software Engineering, pp. 24–35. ACM, New York (2008)
5. Brandes, U.: A Faster Algorithm for Betweenness Centrality. Journal of Mathematical Sociology 25(2), 163–177 (2001)
6. Ford, L.R., Folkerson, D.R.: A simple algorithm for finding maximal network flows and an application to the Hitchcock problem. Canadian Journal of Mathematics 9, 210–218 (1957)
7. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Softw: Pract. Exper. 21(11), 1129–1164 (1991)
8. Hannemann, A., Klamma, R.: Community Dynamics in Open Source Software Projects: Aging and Social Reshaping. In: Petrinja, E., Succi, G., El Ioini, N., Sillitti, A. (eds.) OSS 2013. IFIP AICT, vol. 404, pp. 80–96. Springer, Heidelberg (2013)
9. Howison, J., Crowston, K.: The perils and pitfalls of mining SourceForge. In: Proceedings of the Int. Workshop on Mining Software Repositories (MSR 2004), pp. 7–11 (2004)
10. Howison, J., Inoue, K., Crowston, K.: Social dynamics of free and open source team communications. In: Damiani, E., Fitzgerald, B., Scacchi, W., Scotto, M., Succi, G. (eds.) Open Source Systems. IFIP, vol. 203, pp. 319–330. Springer, Boston (2006)

11. Howison, J., Conklin, M., Crowston, K.: FLOSSmole: A collaborative repository for FLOSS research data and analyses. Int. Journal of Information Technology and Web Engineering 1(3), 17–26 (2006)
12. Kuechler, V., Gilbertson, C., Jensen, C.: Gender Differences in Early Free and Open Source Software Joining Process. In: Hammouda, I., Lundell, B., Mikkonen, T., Scacchi, W. (eds.) OSS 2012. IFIP AICT, vol. 378, pp. 78–93. Springer, Heidelberg (2012)
13. Kunegis, J., Sizov, S., Schwagereit, F., Fay, D.: Diversity dynamics in online networks. In: Proc. of the 23rd ACM Conf. on Hypertext and Social Media, USA (2012)
14. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: Proc. of the SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (2005)
15. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: Proc. of the 17th Int. Conf. on World Wide Web (WWW 2008). ACM (2008)
16. Noack, A.: Energy models for graph clustering. J. Graph Algorithms Appl. 11(2), 453–480 (2007)
17. Nyman, L.: Understanding code forking in open source software. In: Proc. of the 7th Int. Conf. on Open Source Systems Doctoral Consortium, Salvador, Brazil (2011)
18. Nyman, L., Mikkonen, T., Lindman, J., Fougère, M.: Forking: the invisible hand of sustainability in open source software. In: Proc. of SOS 2011: Towards Sustainable Open Source (2011)
19. Robles, G., González-Barahona, J.M.: A comprehensive study of software forks: Dates, reasons and outcomes. In: Hammouda, I., Lundell, B., Mikkonen, T., Scacchi, W. (eds.) OSS 2012. IFIP AICT, vol. 378, pp. 1–14. Springer, Heidelberg (2012)
20. Sowe, S., Stamelos, L., Angelis, L.: Identifying knowledge brokers that yield software engineering knowledge in OSS projects. Information and Software Technology 48, 1025–1033 (2006)
21. Torres, M.R.M., Toral, S.L., Perales, M., Barrero, F.: Analysis of the Core Team Role in Open Source Communities. In: 2011 Int. Conf. on Complex, Intelligent and Software Intensive Systems (CISIS), pp. 109–114. IEEE (2011)
22. Zachary, W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33(4), 452–473 (1977)