Analyzing FOSS Collaboration and Social Dynamics with Temporal Social Networks

Amir Azarbakht, Carlos Jensen

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

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

Keywords. FOSS, Social Dynamics, Temporal Analysis, Free/Open Source Software, FLOSS, Forking, Social Network Analysis

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, as efforts are formed around communities of interest, and depend on continued interest and involvement in order to stay alive [Nyman 2011].

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 raised as the result of divergent opinions about the future of the project might lead to an ensuing fork of project and the 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 [Bezrukova et al. 2010].

Most recent studies of FOSS communities have tended to suffer from two important limitations. First, they treat community as a static structure rather
than a dynamic process. Second, many social dynamics in FOSS have been studied using a case-study methodology, focusing on a selected subset of the available data.

In this paper, we propose to use social network analysis to study the evolution and social dynamics of FOSS communities. With these techniques we hope to identify measures associated with unhealthy group dynamics, e.g. a simmering conflict, as well as early indicators of major events in the lifespan of an online community. One dynamic we are especially interested in are those of forked FOSS projects. We will seek to validate this technique by comparing the results of our analysis to the results of a study of forked FOSS projects by [Robles and Gonzalez-Barahona 2012]. The goal is to demonstrate that this quantitative approach can be applied to commonly available FOSS archives to get a better understanding of the evolution of these communities.

This paper is organized as follows: We present related literature on online social communities, recounting their focus and the findings. We then present the gap in the literature, and what further study needs to be done. Next, we discuss why the issue needs to be addressed and who benefits from it, in the motivation section. Following that, we present three research questions framed as hypotheses that we are going to test. After that, we propose a methodology as to how to test the validity of the hypotheses, which includes gathering data, doing the analysis, and the visualization of the findings. At the end, we present future work and challenges.

2 Related Work

The social structures of FOSS communities have been studied extensively over the past decade. Researchers have studied the social structure and dynamics of team communications [Howison et al. 2006, Bird et al. 2008], identifying knowledge brokers and their associated activities in FOSS projects [Sowe et al. 2006], their sustainability [Nyman 2011], FOSS forking [Nyman 2011, Robles and Gonzalez-Barahona 2012], their topology [Bird et al. 2008], their demographic diversity [Kunegis et al. 2012], gender differences in the process of joining them [Kuechler et al. 2012] and the role of the core team in their communities [Torres et al. 2011]. They 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 on studying information flow and predicting conflict and fission in groups by Zachary [Zachary 1977], to the more recent works of [Leskovec et al.] on the statistical properties of community structure in social networks, there is a growing body of quanti-
tative research on online communities.

The earliest works on communities was done with a focus on information diffusion in a community [Zachary 1977]. 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 [Ford and Fulkerson 1957] 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 projects 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 [Robles and Gonzalez-Barahona 2012] 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. shows their results, discussed further in the next sections, and which we build on extensively in this work. 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. 2009]. 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 to calculate the metrics in Table 1 for the nodes and the network.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>Tendency of a node to have interactions with the same nodes over time</td>
</tr>
<tr>
<td>Sociability</td>
<td>Tendency of a node to have different interactions</td>
</tr>
<tr>
<td>Influence</td>
<td>Number of followers a node has on a network and how its actions are</td>
</tr>
<tr>
<td></td>
<td>copied and/or followed by other nodes. (e.g. when it joins/leaves a</td>
</tr>
<tr>
<td></td>
<td>conversation, many other nodes join/leave the conversation, too)</td>
</tr>
<tr>
<td>Popularity</td>
<td>Number of nodes in a cluster (how crowded a sub-community is)</td>
</tr>
</tbody>
</table>
The communication patterns of FOSS developers in a bug repository were examined by [Howison et al. 2006]. 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. 2012]. Table 2 lists five network statistics they 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. [Kunegis et al. 2012] argued that one possible reason could be that the community structure consolidates as projects mature.

<table>
<thead>
<tr>
<th>Network property</th>
<th>Network is diverse when</th>
<th>A network is diverse when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paths between nodes</td>
<td>Paths are long</td>
<td>Effective diameter</td>
</tr>
<tr>
<td>Degrees of nodes</td>
<td>Degrees are equal</td>
<td>Gini coefficient of the degree distribution</td>
</tr>
<tr>
<td>Communities</td>
<td>Communities have similar sizes</td>
<td>Fractional rank of the adjacency matrix</td>
</tr>
<tr>
<td>Random walks</td>
<td>Random walks have high probability of return</td>
<td>Weighted spectral distribution</td>
</tr>
<tr>
<td>Control of nodes</td>
<td>Nodes are hard to control</td>
<td>Number of driver nodes</td>
</tr>
</tbody>
</table>

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 Motivation

To better understand and measure the evolution and social dynamics of FOSS projects, integral components to understanding their evolution and direction, we need new and better tools. With this knowledge and these tools, we could
help projects reflect on their actions, and help community leaders make informed decisions about possible changes or interventions. We want to map the dynamics of communities to real world phenomena. Identification is the first step to rectify an undesired dynamic before the damage is done. A community that does not manage growing pains may end up stagnating or dissolving.

Managing growing pains is especially important in case of FOSS, where near half the project contributors are volunteers [Forrest et al. 2012]. [Oh et al. 2004] have argued that openness in FOSS is “[…] generally perceived as having a positive connotation, however, the term can also be interpreted as referring to some unconstructive characteristics, such as unobstructed exit, susceptible, vulnerable, fragile, lacking effective regulation, and so on. The unobstructed exit and lack of regulatory force inherent in the OSS community can result in a community’s susceptibility and vulnerability to herded exits by its participants. Commercial vendor intervention, an alternative project becoming available, and licensing issues can result in some original core members ceasing to provide their loyal service for the community, which can prompt their coworkers to leave as well” [Oh et al. 2004].

Recipes for success or stagnation, sustainability or fragmentation could be identifiable, leading to a set of best practices and pitfalls.

4 Research Questions

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 [Robles and Gonzalez-Barahona 2012] classify the main reasons for forking into six classes, listed in Table 3.

<table>
<thead>
<tr>
<th>Reason for forking</th>
<th>Example forks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical (Addition of functionality)</td>
<td>Xpdf &amp; Poppler</td>
</tr>
<tr>
<td>More community-driven development</td>
<td>EGCS &amp; GCC</td>
</tr>
<tr>
<td>Discontinuation of the original project</td>
<td>Apache web server</td>
</tr>
<tr>
<td>Commercial strategy forks</td>
<td>LibreOffice &amp; OpenOffice.org</td>
</tr>
<tr>
<td>Legal issues</td>
<td>X.Org &amp; XFree</td>
</tr>
<tr>
<td>Differences among developer team</td>
<td>OpenBSD &amp; NetBSD</td>
</tr>
</tbody>
</table>

Three of the six listed reasons are socially related, and so should arguably be reflected somehow in the social interaction data. As an example, if a fork occurs
because of a desire for “more community-driven development”, we expect to see an interaction patterns in the collaboration data showing, a strongly-connected core community getting more isolated from the rest of the community, and the formation of another very active and well-connected core. More specifically, our research hypotheses are:

**Hypothesis #1:** Social interactions data can be used to predict an imminent fork.

**Hypothesis #2:** “Personal differences” and “technical differences” are associated with distinctly different interaction patterns. It is possible to tell the difference between when the reason for forking was personal vs. technical.

**Hypothesis #3:** It is possible to tell through collaboration patterns between when the reason for forking is “more community-driven development” and “differences among the developer team”.

5 Proposed Methodology

Determining how FLOSS networks evolve over time requires us to adopt a case study methodology. To this end, we propose to examine several FLOSS projects. This will involve obtaining mailing list archives, bug repositories, Git logs, creating the collaboration graphs, applying social network analysis technique and measuring SNA metrics, and visualizing the evolving graphs. We plan to have five phases as described in the following:

5.1 Phase 1: Data Mining

We plan to study six projects that have forked and six projects that have not forked as a control group. Data collection involves mining mailing list archives, bug repositories, and Git logs. [Howison and Crowston 2004] described their experiences of “spidering, parsing and analysis of SourceForge data.” Building on their research, we will carefully avoid the pitfalls prone to skew the study. We plan to contribute to and use FLOSSmole [Howison et al. 2006] resources, which provides scripts for FLOSS research data and analyses.

5.2 Phase 2: Creating Communication Graphs

Many social structures can be represented as graphs. The nodes represent actors/players and the edges between them represent the interaction among the actors. Such graphs can be a snapshot of the network, which forms a static graph, or a longitudinally changing network, also called a dynamic graph.
In this phase, we process the data to form a communication graph representing the community. This is fraught with complications. Collaboration graph for a large community, e.g. Linux Kernel with 9515 contributors will be a complex network of 9515 nodes and millions of edges. Special graph sampling algorithms are needed to use clustering techniques and social network analysis tools on such a large and complex network. These graph-sampling techniques have been studied and compared by [Wang et al. 2011] and [Lu et al. 2011]. [Chi et al. 2007] e.g. applied a temporal smoothness to their spectral clustering algorithm, and argued that incorporating this method results in a more robust clustering results, while it provides less sensitivity to short-term noise and is “adaptive to long-term clustering drift.”

5.3 Phase 3: Temporal Evolution Analysis

In this phase, we want to analyze the changes that happen to the community over a given period of time, e.g. two years before and two years after the forking event. To this end, we are interested in seeing the changes in the roles individuals have, any shifts in the center of gravity of the project, unusual dilution or concentration of the part of the community and how information can diffuse along the community. These changes can translate into questions, e.g. is everyone still connected to everyone like before, or there are bridges burnt down. The roles individuals take in the community are in par with the individual’s importance or centrality in the network. We can measure each individual’s importance with a social network analysis measure called centrality, i.e. who is more important/central than the others? e.g. who has the most number of friends in a community? (degree centrality)

There are many ways of looking at an individual’s importance. This is measured by a centrality called closeness centrality. The farness of a node is defined as the sum of its distances to all other nodes. The closeness of a node is defined as the inverse of the farness. More informally, the more central a node is the lower its total distance to all other nodes. Closeness centrality can be used as a measure of how fast information will spread through the network [Chakrabarty and Faloutsos 2006]. Secondly, if we are looking for people who can serve as bridges between two distinct communities, we could measure the node’s betweenness centrality. Betweenness centralities for mediators who act as intermediate entities between other nodes are higher [Chakrabarty and Faloutsos 2006]. Third, if cross-community collaboration is the focus, we can measure edge betweenness centrality. Edges connecting nodes from different communities have higher edge centrality values. In the community collaboration graph, edge betweenness or stress of an edge is the number of these shortest paths that the edge belongs to, considering all shortest paths between all pairs of nodes in the graph. Fourth, one can claim that certain people in the community are more important than others, and whoever is close to them, is relatively more
important than others. In graph terms, this is measured by *eigenvector centrality*, which is based on the assumption that connections to high-profile nodes contribute more to the importance of a node. Google’s PageRank link-analysis algorithm by [Brin and Page 1998] is a variant of the eigenvector centrality measure. In short, centrality measures have been used in several studies to identify key player in a community.

In addition to the centrality measures, we plan to look into the *resilience* of the community as well. By resilience, we mean how well the network holds its structure and form when some parts of it are deleted, added, or changed. For a graph, the resilience of a graph is a measure of its robustness to node or edge failures. This could occur for instance when an influential member of the community leaves. Many real-world graphs are resilient to random failures but vulnerable to targeted attacks. Resilience can be related to the *graph diameter*: a graph whose diameter does not increase much on node or edge removal has higher resilience [Chakrabarty and Faloutsos 2006].

![Fig. 1. Heat-map color-coded examples of nodes with high centrality metric are shown above. The same network is analysed four times with the following centrality measures: A) Degree centrality, B) Closeness centrality, C) Betweenness centrality and D) Eigenvector centrality [Rocchini 2012]](image-url)
We plan to measure these metrics defined on the communication graph, as well as clustering coefficient of node, degree distribution of the entire network, and the graph’s diameter, whose definitions are out of the scope of this paper.

The significance of SNA measures on a single collaboration graph is defined as compared to a random graph, e.g. an Erdős-Rényi random graph of the same size [Erdős and Rényi 1959], to examine their relative significance. We plan to do this analysis over time, rather than just a one-snapshot analysis. So, the temporal changes in measured values are what we are primarily interested in.

We are aware that while datasets consisting of huge real-world graphs are obtainable, their sizes may range from thousands nodes and several million edges. While conventional algorithms to compute graph metrics (shortest paths, centrality, betweenness, etc.) exist, many of these algorithms become impractical for large graphs. Leskovec [Leskovec et al. 2008] argues that one can use sampling algorithms – but that is a non-trivial problem since the goal is to maintain structural properties of the network.
5.4 Phase 4: Temporal Visualization

Several visualization techniques and tools are used in the field of social network analysis, for instance Gephi [Bastian et al. 2009], NetLogo [Wilensky 1999], igraph [Csardi and Nepusz 2006], NetworkX [Hagberg et al. 2008], SoNIA [Bender-deMoll and McFarland 2006], NodeXL [Smith et al. 2009]. Gephi, is a FLOSS tool for exploring and manipulating networks. It is capable of handling large networks with more than 20,000 nodes and features several SNA algorithms. It is customizable with plugins and can be used for dynamic network visualization.

5.5 Phase 5: Interviewing & Triangulation

To complement the quantitative techniques, we plan to interview key players identified in the community and the situation studied to gain their perspective on the community as well. This will be useful in cross-validating the results, as the part of communication that was not logged online could be taken into account in this way. Furthermore, we will compare our findings to those of [Robles and Gonzalez-Barahona 2012]. This should allow us to determine how well the SNA techniques work to identify the underlying reasons for a fork.

5.6 Challenges

The proposed technique uses the data from online communications. The assumption that all the communication can be captured by mining repositories is intuitively imperfect, but inevitable. Hence, to minimize the effect of this assumption, we complement the quantitative approach with a qualitative approach of interviewing key individuals from the community. This will help triangulate the results.

6 Preliminary Results

Some initial work is done, namely scraping mailing list data off several FOSS projects, as well as their bug repositories. We have also done a literature review of the existing work on social network analysis on FOSS communities, and have experienced working with large graphs. There are a myriad of possible applications for the possible outcomes of this line of research, and the gap in the existing body of research shows a need for further study of FOSS social network analysis.

7 Acknowledgments

We would like to thank the peers at the Human-Computer Interaction Lab at Oregon State University, with special thanks to Victor Kuechler and Jennifer L. Davidson for their feedback.
References


