Social dynamics of FLOSS team communication across channels

Andrea Wiggins, James Howison and Kevin Crowston
Syracuse University School of Information Studies
Hinds Hall, Syracuse, NY, 13244 USA
{awiggins|jhowison|crowston}@syr.edu,
WWW home page: http://floss.syr.edu

Abstract. This paper extends prior investigation into the social dynamics of free and open source (FLOSS) teams by examining the methodological questions arising from research using social network analysis on open source projects. We evaluate the validity of data sampling by examining dynamics of communication centralization, which vary across multiple communication channels. We also introduce a method for intensity-based smoothing in dynamic social network analysis.

1 Introduction and Literature Review

Increasing use of social network analysis (SNA) techniques in the research of FLOSS development projects raises concern about the validity of the measured constructs. Centralization is a measure of particular interest for describing the organization of FLOSS teams through the structure of social interactions, as FLOSS team structure has often been claimed, by practitioners, to exhibit decentralization, while researchers have argued that communication centralization may indicate the kind of strong leadership that enables success in an otherwise decentralized organizational context. However, centralization can vary widely within teams over time and venues [1-4], so an aggregate network structure may provide an overly simplified representation of the interactions in FLOSS teams. In this paper, we present a dynamic approach to assessing project network centralization. To motivate our proposed approach, we first discuss problems with current approaches, namely inappropriate use of measures, non-dynamic analyses, failure to consider intensity of relationships and a focus on a single forum. We then introduce an alternative approach and illustrate its utility in a study of two FLOSS teams.

Research that applies SNA techniques to studies of FLOSS team communication networks (as opposed to team membership networks [2]), typically constructs a social network by using the reply structure of the public threads in a venue as a proxy for direct communication between individuals. This approach defines a link as the interaction between a replier and the immediately previous poster in a threaded discussion. Studies of the information flow characteristics of social networks assume

that information flows point-to-point, but depending on the mailing list structure, actual message recipients may include all project participants, all previous posters to the thread rather than only the immediately previous poster. The broadcast structure of most of these communication channels therefore restricts the choice of SNA measures that are meaningful, as the potentially public nature of the messages clearly violates the assumptions of standard brokerage measures such as betweenness centralization [5], among others. Following [1], we therefore use outdegree centralization [6] as a whole-network measure of inequality of communicative contributions in the network. High values indicate that a few individuals respond to many more participants, while lower values indicate a more equal sharing of communicative work.

Time also presents several challenges for working with these data; all communication networks are sensitive to validity problems from collapsing events over a long period of time. While aggregating events over time is more analytically tractable, it often masks meaningful dynamics and typically fails to retain information about the intensity of relationships [1,3]. We therefore develop a dynamic analysis of the network. Dynamic analysis requires sampling a time series of snapshots of the networks, based on the time-stamp assigned to the message upon receipt by the message server.

Periods without any communications are surprisingly common, and this presents a further analytical challenge, as a lack of observations does not necessarily equate to a lack of network structure in the community; people may still have on-going relationships even if they haven't spoken for a month. This problem is typically addressed in time-series analysis through smoothing, in which data are divided into overlapping snapshots and sampled in windows (e.g., of 90 days) moving the window forward by a fixed unit (e.g., by 30 days) for each observation [1]. Using a 90-day sliding window, however, means that a single dyad may be reflected in up to three consecutive snapshots; window size is selected to assure that enough observations are present for most forums to generate analysis data for each time period.

A comparison of the effects of smoothing on communication network centralizations is shown in Figure 1; effective smoothing reduces the standard deviation of the network centralizations, but is problematic in that it tends to inflate the mean value. In addition, it tends to "shift" the observations of dynamics forward in time, so that a peak observed in February 2005 with the 30-day window, in Figure 1, only becomes evident in March 2005 with 60-day smoothing, and does not appear until April 2005 when 90-day smoothing is applied.

Intensity of relationships introduces another challenge for SNA in communication networks, for which it has long been known that the strength of ties affects the interactions between individuals [7]. Like most SNA measures, the standard interpretation of outdegree centralization evaluates the centralization of a dichotomous network, those in which ties are either present or absent between any pair of dyads. The measures assume binary relationships because they are designed to evaluate abstract relationships as opposed to the individual communications that

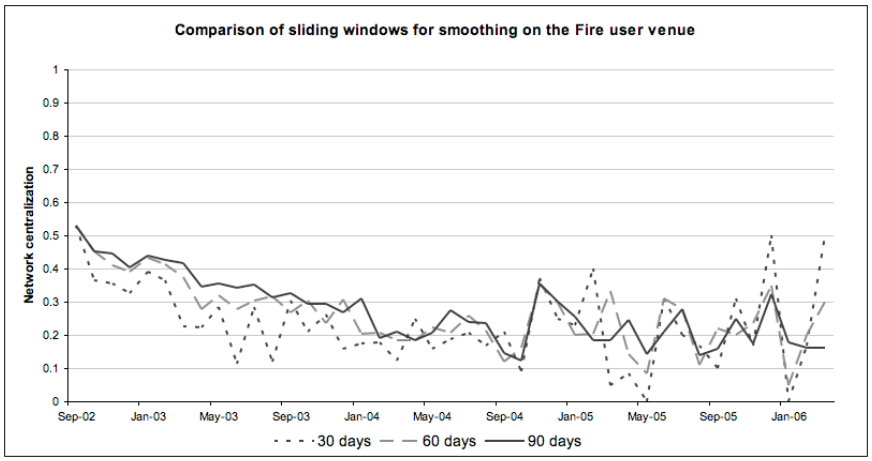

Fig. 1. Comparing the network centralizations observed with different smoothing windows demonstrates the noise reduction from smoothing, such as the period between May and September of 2003. The effect of smoothing also evident for months when no data were available, such as in May 2005, or were very sparse, as in February through April of 2005. For these data, using 90-day periods instead of non-overlapping 30-day periods reduced the standard deviation of the centralizations from 0.14 to 0.12, and increased the mean value from 0.22 to 0.26.

researchers use as a proxy for relationships. In a communication network, a dichotomous representation is a reduction of the available source data.

Most approaches that preserve the information about intensity of interactions in networks employ edge weightings based on unit weighting. Unit weighting increases the weight of each edge by incrementing the edge value by a fixed unit for each message between a pair in the network sample. Node strength is another option for evaluating centrality with this edge weighting method [8], which indicates the volume of activity in dyadic pairs but assumes a fixed value to each interaction. As few robust measures utilize edge weights, the usual compromise dichotomizes networks based on threshold criteria. This allows the analysis of weighted networks using measures that assume dichotomous relationships, but further complicates interpretation by the necessity of selecting threshold criteria, which can be sensitive to such factors as the size of the data sample.

Finally, there are validity considerations inherent in the selection of venues for data sampling. Bug trackers, email lists, and forums have all been used individually for studies of communication networks, but these venues have important differences in function and audience that may affect their interaction dynamics. Many analyses of FLOSS networks analyze only one communication channel to represent the activity of the entire project community. Howison et al. [1] explored the potential for identifying leadership through patterns of contribution to communications, but noted

that development contribution is also considered a strong leadership indicator in FLOSS project teams. Despite theoretical reasons for sampling in particular channels, such as bug trackers or developer email lists, examining the social dynamics of project groups from the perspective of only one communication channel presents an incomplete view of project participation [9]. If we take contribution as a proxy for leadership behavior, we still cannot assume that leadership will be evident in only one of several channels of communication. Participation in bug-fixing, for example, may represent a different form of leadership than participation on a developer or core email list. It is therefore reasonable to expect variance in the communication dynamics of user-oriented and developer-oriented venues, and in discussion-oriented versus bug-fixing venues, which poses a potential threat to convergent validity in FLOSS studies that use SNA methods.

2 Data and Methods

In this section of the paper, we present an intensity-based smoothing method to address challenges with dynamic SNA in communication networks and to mitigate the effects of the overlapping window of observations. In our method the recency of a message affects its salience in an ongoing dynamic structure. From this perspective, a more recent interaction has more impact on the current state of communications in each snapshot than an interaction from the very beginning of the time window selected for analysis.

We then examine the validity of venue selection in the following section by comparing the dynamics of communication in multiple venues within projects to determine whether the centralizations of these communication channels show correlated, comparable patterns of change over the course of the project lifespan. Finally we cautiously interpret these measurements for substantive findings in a comparison between two FLOSS projects.

2.1 Sample Selection and Raw Data

Our analysis focuses on the communication patterns in two projects, Fire and Gaim. These projects are similar in that they are both community initiated multi-protocol instant messaging clients but differ in their ability to sustain project success. Gaim was founded in 1999 as a Linux AOL messenger client and has continued to grow, eventually being ported to all major Operating systems (Windows and Mac OS X). In early 2006, Gaim changed its project name to Pidgin; our data is selected from the period from the founding of the project in November 1999 until the name change in April 2006. Fire was founded in 2001 on Mac OS X and was initially quite successful, but eventually faced difficulties and made its final release in 2006. Our analysis uses the entire range of Fire's active development lifespan, from 2001 through March 2006.

For each of these projects, communications in the form of email lists, forums, and trackers were obtained from the publicly available FLOSSmole [10] and Notre Dame Sourceforge repositories[1], which collected them from SourceForge. These data were imported into a database (now available through FLOSSmole) to allow automated analysis. For the Fire project, the communication channels included two trackers, two developer email lists and one user-oriented email list; for Gaim, the channels included four trackers, a user forum, and two developer email lists.. For the purposes of comparison, we aggregate individual communications channels into audience-based communication venues because these venues support different types of activities (e.g. discussing programming questions versus user support [9]), making it reasonable to expect that the communication patterns will differ for these groupings of venues.

2.2 Operationalization

To implement intensity-based smoothing for communication dynamics, we developed an original exponential decay function that assigns a weight to each interaction based on its recency and then sums the individual interaction weights to find the edge weight for each dyad. The edge weight decay function shown below is calculated using three dates for each event; the beginning (t_1) and end dates (t_n) for the period, and the date of the event (t_e). The function uses a rate parameter r, determined by the recency of the message event within the total time period t, to scale the value of the message weight w: Let $t = t_n - t_1 + 1$ and $t_{elapsed} = t_e - t_1 + 1$ such that $r = (t - t_{elapsed})/t$. Interaction weights w are given by $w = e^{(-ln(t)*r)}$ and the interaction weights for each dyad are summed for the edge weight. This assigns the oldest messages in the period a weight that approaches zero, and messages sent on the final day in the period receive a weight of one.

The exponentially decayed weighting is intended to reduce the effects of overlapping windows for data smoothing. To maintain the analytic value of the outdegree centralization measure, the edge weights were subject to a dichotomization threshold so that only edges with values at or above the 0.8 quantile were used to calculate centralization. This threshold selection was made based on sensitivity analysis on a subset of the data and likely affects the effectiveness of the weighting function; an exhaustive comparison of threshold options is a task for future research. Tests comparing the exponentially decayed weighting compared to unit weighting at this threshold saw no effects when applied to a very large data set, such as the Gaim data. Applied to the Fire venues, however, the exponentially decayed weighting showed some variations from the unit weighting, particularly in the less active developer venues, where the correlation between unit and exponentially decayed edge weightings was only 0.86.

The value of exponential edge weighting is best demonstrated on sparse data, for which it provides better smoothing than absolute dichotomization; the venue that

[1] http://www.nd.edu/~oss/Data/data.html

showed greatest effect from the use of exponential weighting was also the least active overall. In addition, the exponential weighting better reflects sudden changes in levels of activity from period to period which might otherwise be masked by the use of a unit-weighted smoothing window. Preserving this type of dynamic is beneficial when evaluating the changes to network centralizations. Figure 2 shows the differences between this method and the absolute unit dichotomization method for one email list.

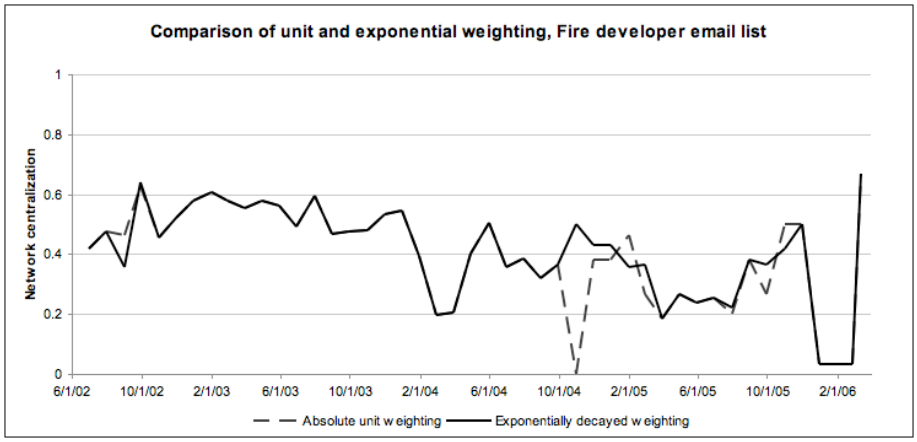

Fig. 2. Comparing network centralizations with absolute unit weighting and exponentially decayed weighting shows no differences for periods of time with higher activity levels, but the exponentially decayed weighting provides better smoothing during periods when there is very little activity.

The dynamic network analysis was performed using a scientific workflow tool, Taverna Workbench, which enabled the development of data analysis workflows that take advantage of modular design and utilize built-in iteration strategies to accomplish a series of data processing tasks over a number of project forum data sets. The workflow used in this analysis was designed to parse mailing list messages into graphs of network centralization over time, depicted in Figure 3[2]. While a brief series of virus messages in one of the venues, identified during content analysis for a separate study, could not be excluded from the current analysis and causes a small effect on one channel, this automated method enabled repeatable analysis of large data sets. For example, the Gaim data set included over 41,000 events in the user forum, over 30,000 events in the developer venues, and about 20,000 events in the trackers.

[2] The workflows and XML records of the workflow runs used to produce the analysis are available at http://ossmole.svn.sourceforge.net/viewvc/ossmole/taverna-workflows/sna/. The workflow is explained in detail in a companion paper proposing a demonstration of the Taverna tool [11].

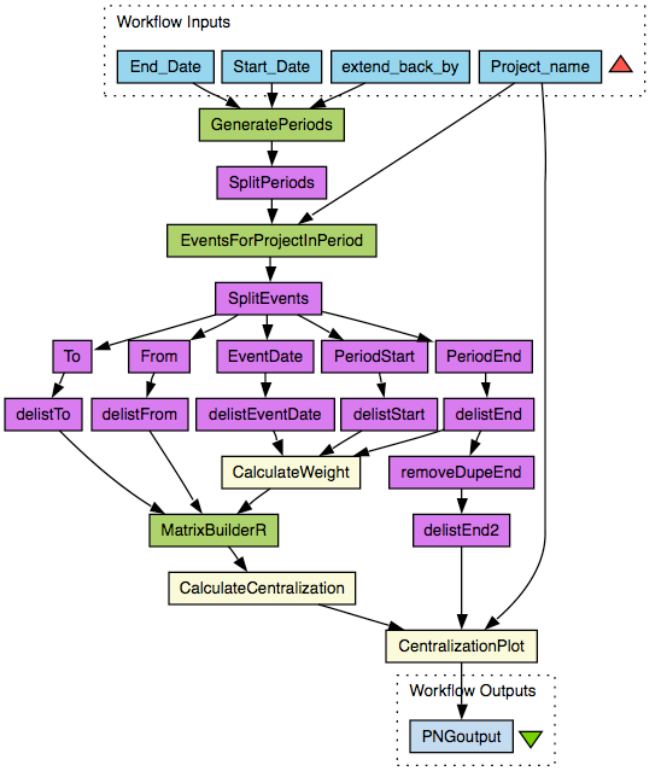

Fig. 3. Taverna Workbench analysis workflow to extract FLOSS message data based on user-provided project name and data sampling time frame for analysis and return an irregular time-series plot of network centralizations. See [11] for details.

3 Communication Venues

To examine communication centralization trends across the venues within each project, and between projects, we compare time series analyses for the two projects. Exponentially decayed smoothing was applied to monthly periods with a 90-day overlapping sliding window for the user, developer and tracker venues in both projects. Each project shows different dynamics in each venue; while both projects tend toward greater decentralization in communications over time, they display varying patterns of interaction. This variation is evident in the correlations between the communication venues within each project.

In three venues for the Fire project (Figure 4) there are comparable mean values for network centralization of trackers and developer email lists (Figure 5). The user email list had a lower average centralization, which reflects the larger and more

diverse group of message respondents. The standard deviations of the centralizations are similar for the user and developer venues but higher for trackers due to a spike in centralization values in December of 2005, which affects the values for the following two months due to the sliding window. The sudden change from a very decentralized structure to a highly centralized structure in December of 2005 originates in the feature requests tracker, when one individual closed 279 bugs in a very short period of time. This was most likely in preparation for the end of project development activity, as the final release of Fire followed three months later.

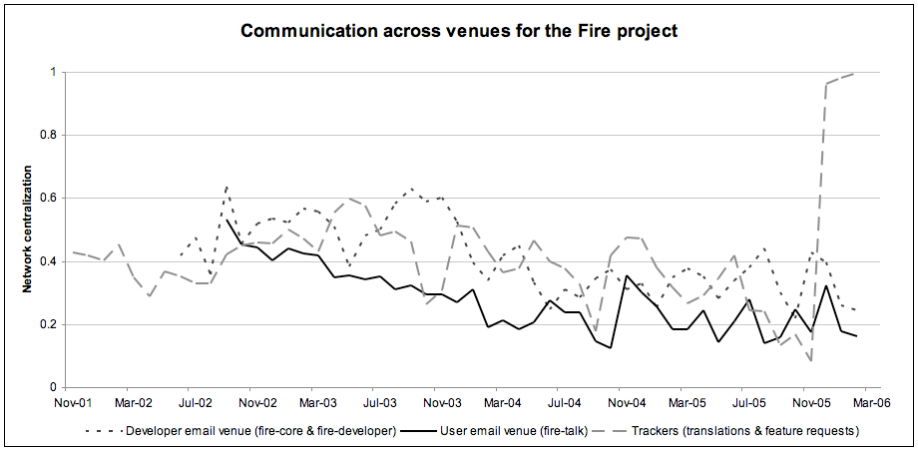

Fig. 4. Communication networks in different venues for the Fire project showed different dynamics over time, although all of the venues show a trend toward increasingly decentralized communications. The tracker shows an interesting exception to this trend just prior to the end of the project activity.

Excluding this period of unusually high centralizations, the mean and standard deviations of the tracker centralizations are comparable to those for the email lists, shown in Figure 5. This may imply some level of regularity across these different venues, and there is an overall downward trend in each venue as communications become increasingly decentralized. Despite these similarities, the three Fire communication venues clearly display different dynamics over time; the correlations in Table 1 show that the developer and user venues are most similar, while the developer venue and the trackers were negatively correlated.

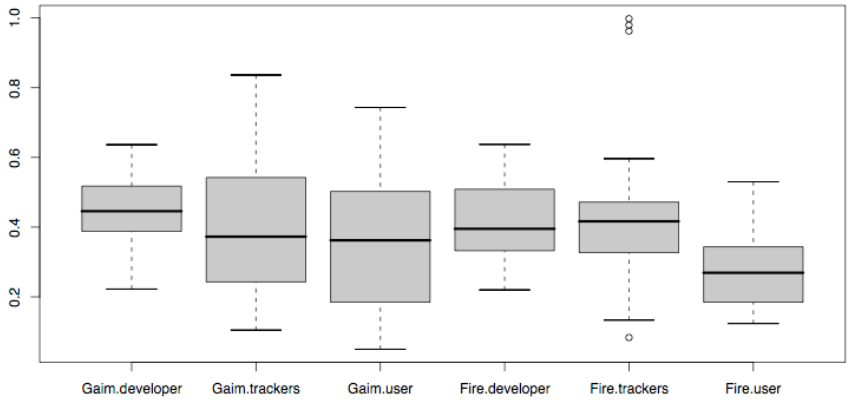

Fig. 5. The distributions for the Gaim and Fire network centralizations show similarity in the summary statistics for the trackers and user venue in Gaim that do not appear in Fire. For each project the user venues have lower centralizations than the developer venues.

Gaim also shows different communication dynamics in different venues (Figure 6); the average centralizations are lowest for the user forum and highest for the developer list, which has a smaller number of participants. The standard deviations of the centralizations for the user forum and the tracker are comparable, while the standard deviation for the developer list was much lower, and visual inspection of the centralization trends reflects a more varied participation dynamic in the user forum and trackers. Periodic spikes in tracker activity appear to indicate project "housecleaning" much like the phenomenon observed at the end of the Fire project, as there were several periods during which a large number of bugs were closed. If these large batches of bug closing were conducted by one (or very few) individuals, as appears to be the case, this would generate the observed highly centralized network structures.

Table 1. Correlations of centralizations between communication venues in Fire and Gaim highlight differences in the project communication dynamics.

Project	User-Developer	Developer-Trackers	User-Trackers
Fire	0.62	-0.03	0.21
Gaim	0.17	0.57	0.57

Gaim's user and developer venues both bear greater similarity to the trackers than to one another (Table 1). This is very different from the correlations for Fire, where the user and developer venues are most alike, and may indicate different uses of these venues by the project participants. One possible explanation for this

difference between the projects is that both the user and developer venues were email lists for the Fire project, while the user venue in Gaim was a forum. However, this would not explain the very different correlations between activity in developer venues and trackers, or the similarity of the Gaim trackers to the user and developer venues.

The Gaim venues also show a trend toward decreasing centralization of communications, but the end of the data sample appears to show a more stable range of centralization values. This is confirmed by lower standard deviations for the user and tracker venues, both shifting from approximately 0.18 to 0.9 during the final two years, suggesting that more stable communication patterns have emerged in these venues as the projects matured. At the same time, the developer venue shows little change to standard deviations throughout the project lifespan, which may indicate a different strategy for moderation of development activities over time.

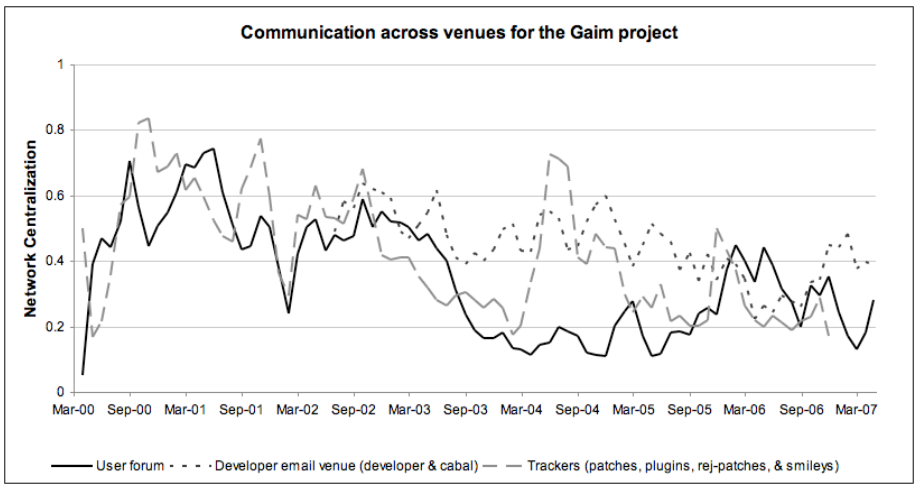

Fig. 6. Communication networks in different venues for the Gaim project also display different dynamics over time; the centralizations for both the user and developer venues were more strongly correlated with the trackers than with one another.

4 Discussion

The use of an intensity-based smoothing assists most with smoothing sparse data; however, the necessity for dichotomization introduces another complication with respect to sensitivity to threshold values, and loses much of the information on the strength of relationships. The selection of smoothing window length may also play a role in the effectiveness of effectiveness of the exponentially decayed weighting, and is another topic for future research.

Analysis of the dynamics across venues shows different levels of correlation between venues for each project, suggesting that these different communication channels may be proxies for different types of relationships. In both projects, the user venue is more decentralized than the developer venue, reflecting the greater number of participants. Another common feature for all venues in both projects was the overall trend toward decentralization over time, although this could be the result of different influences in each case. In the Fire project, decentralization may be the result of loss of project leadership, while in the successful Gaim project it appears to reflect growth in user participation. As a whole, the variation in communication dynamics suggests that convergent validity is an important consideration for studies of FLOSS communication networks, and care should be taken in the selection of venues for data sampling, as observations in different venues will generate different results.

In addition, an interesting phenomenon was observed in each of the projects' trackers, wherein periodic mass bug closings by very few individuals caused sudden, isolated spikes in centralization values. This apparent "housekeeping" behavior, occurring several years into both of these projects, may be a common practice in managing resources for a long-term FLOSS project. We hope to continue the analysis with a larger number of projects to determine whether this phenomenon is common to other projects, which would pose additional challenges to validity for using bug trackers as a data source for analysis of communication dynamics, but which would also point to rhythms in group work, believed to be important to success in distributed environments [12].

5 Conclusion

The dynamic analysis of FLOSS team communications across channels has provided these findings:

- Communication centralization dynamics vary in different venues, suggesting that communication in these venues may be proxies for different kinds of relationships and that researchers should be cautious in using individual venues to characterize projects.
- Periodic project management activities in the trackers were evident in both projects as batch bug closings by a few individuals caused a sudden, temporary shift to a highly centralized network structure. This is both an interesting behavioral phenomenon and a potential confound to analysis based on bug trackers.
- All venues in both projects tended toward decentralization over time, a pattern we expect to observe in future analysis of additional projects. Periods in which centralization bucks this trend and rises might be particularly interesting for further study.

This paper also contributes an original method for computing exponentially decayed edge weightings in a dynamic network and makes it available to the

research community. Future research could extend this work by examining alternate measures of centrality, and by comparing the individual centralities of developers in multiple channels for each project over time, examining both the dynamics of the individual and aggregated communication channels.

References

1. Howison, J., K. Inoue, and K. Crowston. *Social dynamics of free and open source team communications*. in *IFIP 2nd International Conference on Open Source Software*. 2006. Lake Como, Italy: Springer.
2. Gao, Y. and G. Madey, *Network Analysis of the SourceForge.net Community*, in *The Third International Conference on Open Source Systems (OSS 2007), IFIP WG 2.13*. 2007: Limerick, Ireland.
3. Braha, D. and Y. Bar-Yam, *From Centrality to Temporary Fame: Dynamic Centrality in Complex Networks.* Complexity, 2006. **12**: p. 59-63.
4. Crowston, K. and J. Howison, *The Social Structure of Free and Open Source Software Development.* First Monday, 2005. **10**(2).
5. Wasserman, S. and K. Faust, *Social network analysis: methods and applications*. Structural analysis in the social sciences ; 8. 1994, Cambridge: Cambridge University Press. xxxi, 825 p.
6. Freeman, L., D. Roeder, and R. Mullholland, *Centrality in Social Networks: Ii. experimental results*. Social Networks, 1980. **2**: p. 119-141.
7. Granovetter, M.S., *The Strength of Weak Ties*. The American Journal of Sociology, 1973. **78**(6): p. 1360-1380.
8. Valverde, S., et al., *Self-organization patterns in wasp and open source communities.* IEEE Intelligent Systems, 2006. **21**(2): p. 36-40.
9. Sowe, S.K., I. Stamelos, and A. Lefteris, *Identifying knowledge brokers that yield software engineering knowledge.* Information and Software Technology, 2006. **48**(11): p. 1025-1033.
10. Howison, J., M. Conklin, and K. Crowston, *Flossmole: A collaborative repository for FLOSS research data and analysis.* International Journal of Information Technology and Web Engineering, 2006. **1**(3): p. 17-26.
11. Howison, J., A. Wiggins, and K. Crowston, *eResearch workflows for studying free and open source software development*, in *Submitted to Fourth International Conference on Open Source Software (IFIP 2.13)*. 2008.
12. Maznevski, M.L. and K.M. Chudoba, *Bridging space over time: Global virtual team dynamics and effectiveness.* Organization Science, 2000. **11**(5): p. 473-492.