Analyzing leadership dynamics in distributed group communication

Kevin Crowston
School of Information Studies
Syracuse University
Syracuse, NY 13244
Email: crowston@syr.edu

Andrea Wiggins
School of Information Studies
Syracuse University
Syracuse, NY 13244
Email: awiggins@syr.edu

James Howison
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
Email: jhowison@cs.cmu.edu

***Abstract*—We apply social network analysis (SNA) to examine the dynamics of leadership in distributed groups, specifically Free/Libre Open Source Software development projects, and its relation to group performance. Based on prior work on leadership in distributed groups, we identify leaders with those who make the highest level of contribution to the group and assess the degree of leadership by measuring centralization of communications. We compare the dynamics of leadership in two FLOSS projects, one more and one less effective. We find that in both projects, centralization was higher in developer-oriented communications venues than in user-oriented venues, suggesting higher degrees of leadership in developer venues. However, we do not find a consistent relation between centralization and effectiveness. We suggest that SNA can instead be useful for identifying interesting periods in the history of the project, e.g., periods where the leadership of the project is in transition.**

I. Introduction

Distributed groups are groups of geographically dispersed individuals working together over time towards a common goal. Though distributed work has a long history (e.g., [1, p. 165]), advances in information and communication technologies have been crucial enablers for recent developments of this organizational form [2] and as a strategic solution to organizational needs, distributed groups are becoming more popular [3]. While distributed groups have many potential benefits, distributed workers face many real challenges. Watson-Manheim et al. [4] suggest that distributed work is characterized by numerous discontinuities: a lack of coherence in some aspects of the work setting (e.g., organizational membership, business function, task, language or culture) that hinders members in making sense of the task and of communications from others [5], or that produces unintended information filtering [6] or misunderstandings [7]. As a result, more effort is required for interaction when participants are physically distant and unfamiliar with each others' work.

Distributed groups seem particularly attractive for software development because the code can be shared via the same systems used to support group interactions [8]. However, the issues identified above are particularly problematic for software developers [5]. Numerous studies of the social aspects of software development groups [5], [9], [10] conclude that large system development requires knowledge from many domains, which is thinly spread among different developers [9]. As a result, large projects require a high degree of knowledge integration and the coordinated efforts of multiple developers [11]. The additional effort required for interaction often translates into delays in software release compared to traditional face-to-face groups [12], [13]. The problems facing distributed software development groups are reflected in Conway's law, which states that the structure of a product mirrors the structure of the organization that creates it [14], [15]. Accordingly, splitting software development across a distributed group would be expected to make it hard to achieve an integrated product [15].

In response to the problems created by discontinuities, studies of distributed groups stress the need for a significant amount of time spent learning how to communicate, interact and socialize using computer tools [16]. In effective distributed groups, members must share knowledge and information, and create new practices to meet the task-related and social needs of the members [17]. Researchers have further suggested that group leadership helps group members overcome barriers to performance [18]–[20]. However, the nature of leadership in distributed groups does not seem to be adequately described by current theories of leadership that were largely designed to address the interaction between subordinates and a single individual who occupies a formal, appointed managerial/supervisory position in a hierarchical organizational setting [21], motivating a more detailed examination of the functioning of leadership in these settings.

The goal of this paper is to use SNA techniques to analyze leadership in a particular distributed setting, namely among programmers developing Free/Libre Open Source Software (FLOSS). Key to our interest is the fact that most FLOSS projects are developed by distributed groups comprising professionals and users [22], [23]. These groups are close to purely virtual in that developers contribute from around the world, meet face-to-face infrequently if at all and coordinate their activity primarily by means of a variety of computer-mediated communication (CMC) tools [24], [25]. As a result, FLOSS projects provide an interesting setting in which to study distributed group dynamics. Based on a review of the literature, we develop a description of the dynamics of leadership in FLOSS projects and examine the connection between group leadership and overall group effectiveness.

978-0-7695-3869-3/10 $26.00 © 2010 IEEE

The remainder of the paper is organized as follows. First, we review past studies on leadership in distributed groups, followed by those that have applied SNA to FLOSS settings. Next, we present an empirical study that compares a dynamic analysis of FLOSS group communication across multiple venues for pair of projects differentiated by their degree of effectiveness. We conclude with some suggestions for future work using these techniques.

II. Theory development

In this section we present a more specific set of research hypotheses for our study. The hypotheses are based on a theoretical perspective on FLOSS projects developed by reviewing prior studies, starting with discussion of the problems of leadership faced by distributed groups, followed by a discussion of prior applications of SNA to FLOSS projects in particular.

A. Interaction and leadership in distributed groups

As noted above, FLOSS projects face particular challenges to effective work because of their highly virtual nature. While research has suggested that group leadership helps group members overcome barriers to performance, FLOSS projects often work outside the boundaries of any organization, meaning they rarely have formally appointed leaders. In the absence of formal leaders, members within the group lead on a "voluntary" basis, either individually or collectively. In these circumstances, leadership is said to be emergent. According to Berdahl and Craig [26], leaders emerge when "one or more of a group composed initially of equal status peers... exhibits notably higher levels of leadership behavior and thereby attains higher status in the eyes of fellow group members" (p. 26).

Some distributed groups evolve a leadership structure in which a single member emerges who is recognized by other members as the group's leader, while other groups will evolve a less-centralized leadership structure based on interaction and influence patterns. In the latter case, leadership can be shared among two or more group members or distributed among many group members [21], [27], presenting a very different form of leadership. Indeed, in many FLOSS projects, no individual holds formal authority and there may be no single dominant contributor. When asked who their leaders are, members of these groups will often say, "We have no leaders." But if members of a group claim to have no leaders, is it accurate to say that the group has no leadership?

We argue that leadership may be distributed among the members of the group, and that examination of their patterns of communications can help to distinguish the nature of leadership in such groups. Two studies have examined the relationship between emergent leadership and initiation of communication. Both suggest that taking initiative is positively associated with being identified as an emergent leader [28], [29]. This finding applies to the initiation of communication at the outset of a group's task, as well as to initiating communication that keeps members focused on the task throughout the group's life [28]–[30]. As well, findings from several studies of distributed group dynamics suggest that emergent leaders communicate with group members more frequently than non-leaders [29]–[34]. These findings regarding the pattern of communications motivate the application of SNA techniques to this problem. Specifically, based on these findings, we will identify leaders of projects as those who contribute to the project at a higher level, initiating more communications. Furthermore, we expect the emergence of such leaders to be associated with higher levels of group effectiveness, as leaders help the group overcome barriers to performance.

B. SNA studies of FLOSS projects

We turn next to a review of research that has used SNA to examine FLOSS development groups. FLOSS research using SNA has examined two distinct types of network structures: association networks, and communication networks. Association networks are bipartite, having nodes representing individuals and nodes representing entities through which the individuals are associated. FLOSS association networks are most often based upon developer membership in multiple projects (e.g., [35]–[37]), though they can also be based on interactions through code units or participation in mailing list threads [38]. Using such an approach, Christley and Madey [39] identified a set of social positions in the SourceForge community (e.g., user, developer, bug reporter) and showed how these positions evolved over time. The bipartite network can be converted to a unipartite network by dropping one type of node and directly linking the nodes it connects (e.g., linking together all developers who work on the same projects). López-Fernández et al. [40] used this approach to develop networks of modules and of developers for the Apache httpd, KDE and GNOME projects, allowing quantitative comparisons between them.

Another approach, and the one adopted in this paper, develops a unipartite network of individuals with links based on observed interactions. These networks are often developed as communication networks, connecting developers based on who has communicated with whom, which is most relevant for the theoretical perspective developed above. Data for such networks come from email list archives or other developer communications. Note that an association network (or its unipartite projection) is non-directional, while communication networks can be directional (e.g., if a link represents sending an email message). Crowston et al. [41] analyzed communications networks to identify core and peripheral members of FLOSS projects using an SNA core-and-periphery analysis [42], which divides the network between a highly interconnected core and a periphery connected to the core only. They also identified the core group based on level of participation. They found good agreement between these two approaches and that most projects had a very small core of developers. This work suggests that the core members of the project are those who contribute the most, consistent with the findings of the behaviours of group leaders. However, that work examined only a single specialized communication venue, namely bug trackers.

III. Operationalizing leadership with network data

In this section, we discuss how network data, specifically the distribution of levels of communication, can be used to operationalize the concepts of leadership and its relation to the overall effectiveness of the projects. We pay particular attention to the assumptions underlying SNA measures in the FLOSS setting. While network analysis methods are evolving at a rapid pace, several common assumptions of social network analysis affect interpretations of changing networks. The methods for acquiring data for network construction and the observability of relationships in the network influence the way we understand measures of information flow and importance. In addition, we suggest that the expected stability of the constructs of interest and the time dimension of the network data can affect the interpretation of SNA results.

A. Network data and analysis assumptions

First, a matter of continual concern in network studies is the ability to harvest or construct complete, authoritative networks while dealing with the validity issues that arise in data collection [43]. Roughly speaking, two approaches to data collection can be distinguished: those that rely on reports of relationships and those that infer them from archival data. On the one hand, early network analysis is often credited to roots in sociometry, where psychology studies were based on self-reported relationships elicited in surveys or interviews [44], [45]. On the other hand, the use of archival data has a long history, being used for studies of board interlock analysis as early as 1914 [46], [47], providing a long history of archival data mining for indirect evidence of social networks. Despite potential problems with the quality of archival records, these data are considered more reliable than data elicited directly from participants, which may have stronger validity, but is often subject to limitations of recall [48]. In our case, we had access to archives of developer interactions in the form of mailing list archives, but anticipated a low response rate to any survey that might be attempted, which would make construction of a complete network difficult if not impossible. We therefore chose to base our analysis on the archival interaction data.

For this study, we were interested in assessing individual contributions to the group that indicate group leadership. Accordingly, we examined contributions of individuals to group communications venues, as these are where individual group members can influence each other and thus the project outcomes. Given this choice, we still needed to select the specific communications from which to collect data. Issue trackers, email lists and discussion fora have all been used individually for studies of communication networks. However, these venues have important differences in function and audience that may affect their interaction dynamics, Participation in bug-fixing, for example, may represent a different form of leadership than participation in discussion on a developer or core email list, as seen in analysis of the negotiation processes of bug-fixing [49]. Howison et al. [50] 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 groups. Therefore, examining the social dynamics of project groups from the perspective of only one communication channel may present an incomplete view of project participation [51]. To address this concern, we compared across these various communications venues, and expected to find variance in the communication dynamics of user-oriented and developer-oriented venues and in discussion-oriented versus bug-fixing venues.

B. Network Conceptualization

Second, in order to be able to apply SNA techniques, researchers must identify nodes and links. A key problem in using archival data is establishing a connection between recorded behaviours and these theoretical constructs of interest, since the archival data are not typically collected for scientific purposes but rather exist as a by-product of the work in a distributed group. As in many studies, we are studying networks of individuals, who constitute the nodes in the network. As well, we wanted to develop the network from the observed communications among individuals. Nevertheless, we still faced many choices for interpreting the data to identify links between individuals. Unfortunately, many studies are vague about the theoretical rationale for the choice of a particular operationalization of a link.

In our case, the particular communications channels being analyzed are broadcast: messages sent on a mailing list or to discussion forum are available to all members at once. We therefore could not identify links based on the destination of messages, as would be possible with individually addressed messages. Instead, as is common in FLOSS studies based on developer interactions, we identified links from the replies to posted messages, such that developer A is linked to B if A has replied to a message sent by B (e.g., [52]–[54]). This kind of interaction is still theoretically relevant, as someone who is replied to more often initiates more of a discussion while someone who replies more often makes a larger contribution to the group, both factors related to leadership in prior studies. However, we note below some significant limitations to interpreting these interactions.

A further complication in analysis is the choice between a weighted and unweighted network. In a weighted network, the strength of links varies, e.g., depending on the frequency of interactions, while in an unweighted network, links are binary, either present or absent. With archival data, it may be feasible to count interactions, allowing the creation of a weighted network. In principle this is desirable, as a weighted network provides more information. Unfortunately, many network analysis measures were designed for unweighted networks and either do not work or have to be adapted for use with weighted networks. It is therefore common to dichotomize weighted networks, discarding all links below a certain threshold. Because we were working with archival data, we initially created a weighted network, but dichotomized the network for further analysis. Only edges with values at or above the 0.8 quantile of all edge values were used to calculate centralization (i.e., the top 20% of links). This threshold selection was made based on sensitivity analysis on a subset of the data; an exhaustive comparison of threshold options is a task for future research.

C. Network analysis measures

A third issue is the choice of analysis measures on the networks developed. In many SNA studies, measures of individual member importance or influence such as individual centrality are the focus of the analysis. In our case though, we are interested in assessing the overall pattern of communication in the projects. We therefore examined network centralization, which assesses the degree of inequality in individual centralities. In a perfectly centralized network, a single individual has high centrality, while in a perfectly decentralized network, all members are equal. Therefore, a high level of centralization can be interpreted as indicating the presence of a strong leader or leaders, and low centralization as their absence. Centralization is of particular interest for describing the organization of FLOSS projects, which have often been claimed by practitioners to be quite decentralized, with no formal leadership. On the other hand, researchers have argued that communication centralization may indicate the kind of strong leadership that enables success in an otherwise decentralized context, as noted above [18]–[20].

A complication is that there are several different measures to assess individual centrality and thus network centralization, including degree, closeness, and betweenness centrality [55] among other measures of prestige and importance in networks [43]. In choosing between measures, it is important to consider the assumptions about the nature of the underlying social process needed to interpret the measure. For example, interpretation of betweenness centrality as a reflection of importance assumes that a node between other nodes has some influence on the flow between the other nodes, thus increasing the between node's importance. However, in our case, links reflect replies to messages rather than information flow. Therefore, we had to avoid measures of centrality that assumed individual control over information flow. Instead, following [50], we selected out-degree centralization [56] as a whole-network measure of inequality of communicative contributions in the network that is suited to both the data and construct of interest.

We note though that using out-degree centrality to compare the individual actors in the networks simply shows which actors talked the most during any given time period (for a weighted network) or had the most communication partners (for an unweighted network). Were this the extent of our analysis, we would not need to use SNA measures at all, as there are far easier ways to assess volume of individual contributions than creating and analyzing network structures.

D. Network dynamics

Finally, consideration of the dynamics of networks over time is important for meaningful interpretations of SNA measures, and as more longitudinal behavioral data are available this becomes an increasingly salient consideration. Centralization can vary widely within groups over time and across venues [37], [40], [50], [57], [58]. Two relevant dimensions of network data include the temporal nature of the data and the stability of the construct of interest, as shown in Table I. In many SNA studies, particularly those using survey data, data on the network is collected at a single point in time, forming a snapshot view of the network structure. On the other hand, when networks are derived from archival data, the underlying data are typically a time series of events (e.g., email messages), collected across some period of time. These data must then be aggregated to reveal the overall network structure (e.g., deriving a communications network from a year's worth of email exchanges). Turning to the theoretical concerns, the particular construct of interest operationalized through network measures may be conceptualized as being stable (e.g., long-term friendship ties) or dynamic, in which case the network structure evolves over time [59], [60].

TABLE I
TIME CHARACTERISTICS OF DATA AND STABILITY OF CONSTRUCT DETERMINE APPROPRIATENESS OF AGGREGATE OR DYNAMIC ANALYSIS.

Network data characteristics	**Stable Construct**	**Unstable Construct**
Snapshot data	aggregate	aggregate
Longitudinal data	aggregate	dynamic

For most pairings of data type and construct stability, aggregating the network structure data does not significantly impact the validity of analysis and interpretations, with the exception of the longitudinal-dynamic pairing. For example, if friendship is thought to be a stable construct, a friendship network could be derived equally from survey data or longitudinal archival data. On the other hand, for a construct like leadership that is expected to vary, these two approaches might give quite different results. Thorough investigation would be needed to verify that a longitudinal network data set was in fact stable with respect to the construct of interest before applying aggregate analysis measures. Given the nature of our data and our interest in the dynamics of networks, we determined the need to avoid overly aggregating our data and therefore analyzed our measures dynamically as time series.

To summarize this discussion, our research hypotheses can be restated in terms of the SNA measures and their comparison. First, we expect the presence of leadership to be reflected in higher levels of centralization for a project, and to see these higher levels in developer venues more specifically. We also expect higher levels of centralization, reflecting leadership, to be associated with more effective projects. Finally, we will examine the pattern of centralization over time to address the question of the stability of leadership in our projects. If the level is stable, then future studies can adopt a simpler strategy of analyzing aggregated data, but if there is significant variation over time, then a dynamic analysis will be recommended.

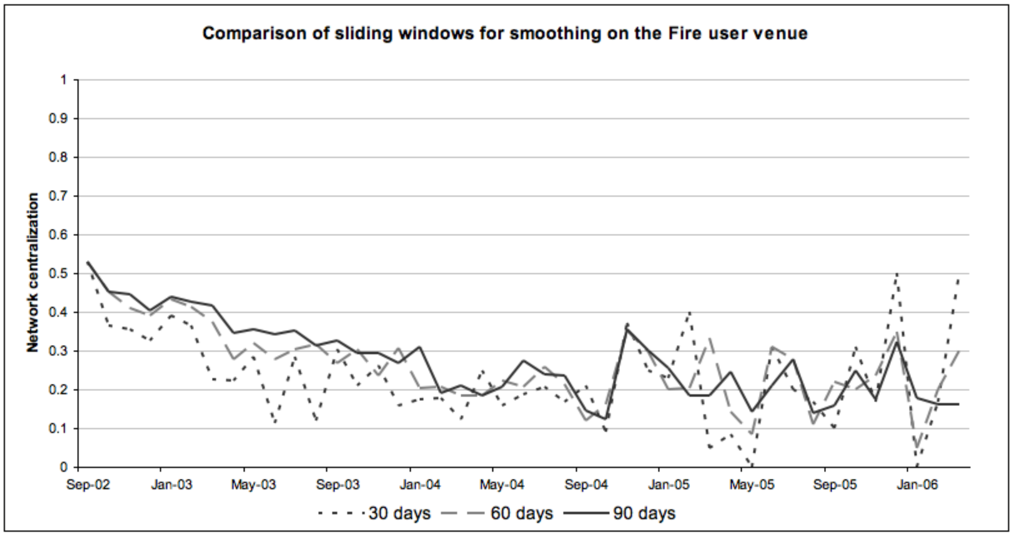

Fig. 1. Comparison of noise reduction from smoothing with different smoothing windows.

IV. Data and Analysis

In this section of the paper, we discuss the design and execution of our study. We first discuss the selection of projects for analysis and the extraction of network data for these projects. We then discuss how these data were analyzed.

A. *Sample Selection*

Our analysis examines the communication patterns in two FLOSS development projects, Fire and Gaim. These projects are similar in that they are both community-based projects developing multi-protocol instant messaging (IM) clients, but differ in their ability to sustain project effectiveness. Gaim was founded in 1999 as a Linux AOL messenger client and has continued to grow, eventually being ported to Windows and Mac OS X. In early 2006, Gaim changed its project name to Pidgin, and continues to be a effective growing project in 2009; 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 effective, but eventually faced difficulties in sustaining development and made its final release in 2006. Our analysis uses the entire range of Fire's active development lifespan, from 2001 through March 2006. A comparison of these two projects, one obviously effective and the other less so, can provide some suggestions as to the connection between communications dynamics and overall project effectiveness, though there are clear limits to the strength of conclusions that can be drawn from two projects.

B. *Data*

For each of these projects, a diverse set of communications in the form of email lists, forums, and trackers were obtained from the FLOSSmole [61] and SRDA repositories [62], [63], which acquired them from SourceForge. 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 aggregated these individual communications channels into three audience-based communication venues: developer, tracker and user. These venues support different types of activities (e.g., discussing programming questions versus user support [51]), making it reasonable to expect that the communication patterns will differ for these groupings of venues based on their functional purposes.

These data were imported into a database to allow automated analysis. The Fire data set includes about 1,800 events in the user email list, 7,800 messages in the developer venues, and 1,300 events in the combined trackers, spanning a period of 54 months. The significantly larger 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, generated over 78 months. A brief series of virus messages in one of the channels, identified during content analysis for a separate study, could not be excluded from the current analysis and caused a minor effect in one communication venue, but without affecting our overall conclusions.

C. *Analysis methods*

In this section, we discuss the methodology we applied to assess the dynamics of social networks in the two FLOSS projects. As noted above, time presents challenges for working with these data. As we previously argued, the dynamic nature of the data on which communication networks are based makes them sensitive to validity problems from collapsing events that occur over a long period of time. We therefore developed a dynamic analysis of the network, sampling a time series of snapshots of the networks based on the time-stamp assigned to the message upon receipt by the message server.

One complication we encountered was that in the venues for the projects we studied, periods without any communications were surprisingly common. However, a lack of observations

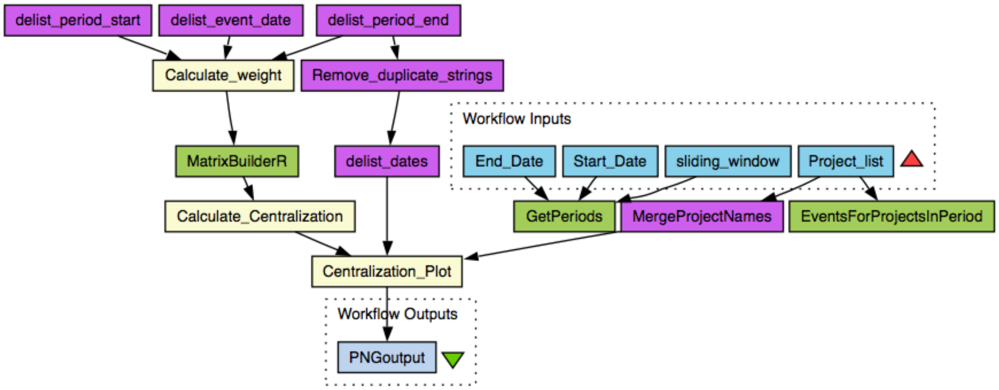

Fig. 2. Taverna Workbench analysis to create time series of network centralizations.

does not necessarily equate to a lack of network structure in the community, as participants may have ongoing relationships even without contact during a particular month. We addressed this problem 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 [40]. The window size was selected to assure that enough observations are present to generate analysis data for each time period. After examination of the temporal pattern of our data, we settled on analysis based on monthly periods with a 90-day overlapping sliding window.

We note though that this approach results in a single observed communications dyad being reflected in up to three consecutive monthly averages. 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 for use with descriptive statistics in that it tends to inflate the mean value. In addition, smoothing shifts the observations of dynamics forward in time, so that a peak observed in February 2005 with the 30-day window (with no smoothing), in Figure 1, only becomes evident in March 2005 with 60-day smoothing, and does not appear until April 2005 with 90-day smoothing. Therefore, it may be necessary to trace the source of sudden shifts in centralization backward to events occurring prior to the apparent manifestation.

D. Research tools

The dynamic network analysis was performed using a scientific workflow tool, Taverna Workbench [64], 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 communication venue 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 2. The use of this automated method enabled repeatable analysis of large data sets, such as that required for the analysis of the sensitivity of the data to the dichotomization threshold.

V. Findings

In this section we present our findings regarding the centralization of communications in the two projects, deferring discussion of the implications of these findings.

Dynamics of centralization. Our first finding is that the centralization of the communications in the difference venues varied quite considerably. Figure 3 shows the time series for the Fire project and Figure 4, for Gaim. Figure 5 shows the distribution of centralizations for the projects. It is clear that there is considerable variation in centralization, suggesting that an aggregated analysis would be inappropriate. Furthermore, each project shows different dynamics in each venue, again illustrating the value of a dynamic analysis.

Centralization across venues. To examine communication centralization trends across the venues within each project, and between projects, we compared time series analyses for the two projects. In three venues for the Fire project (Figure 3) there were 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 were similar for the user and developer venues but higher for trackers due to a spike in centralization values in December of 2005, which affected the values for the following two months due to the sliding window. This sudden change from a very decentralized structure to a highly centralized structure originated in the feature requests tracker, when one individual closed 279 tracker items in a very short period of time. This housekeeping was most likely in preparation for the end of project development activity, as the final release of Fire followed three months later. Excluding this period of unusually high centralizations, the mean and standard deviations of the tracker centralizations were comparable to those for the email lists, shown in Figure 5. This comparability implies some level of regularity in leadership across these different venues.

Gaim also shows different communication dynamics in different venues (Figure 4); the average centralizations are lowest for the user venue and highest for the developer list, which

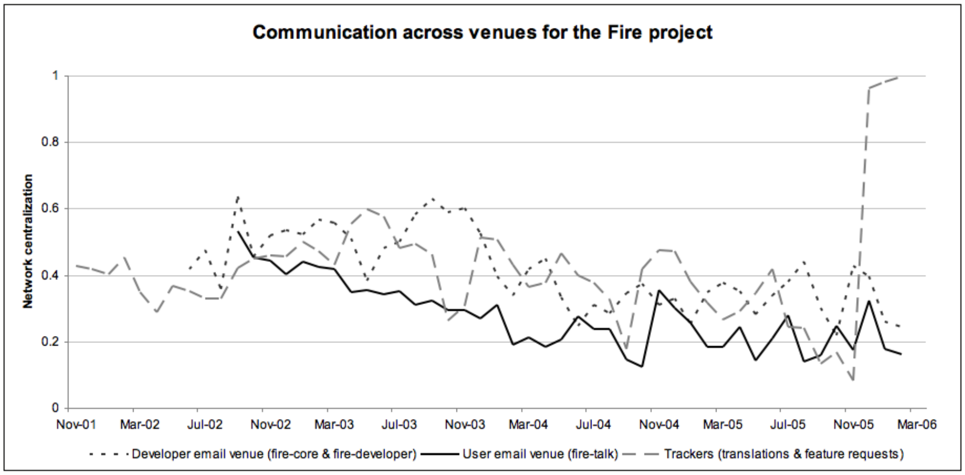

Fig. 3. Dynamics of communication networks in multiple venues for the Fire project.

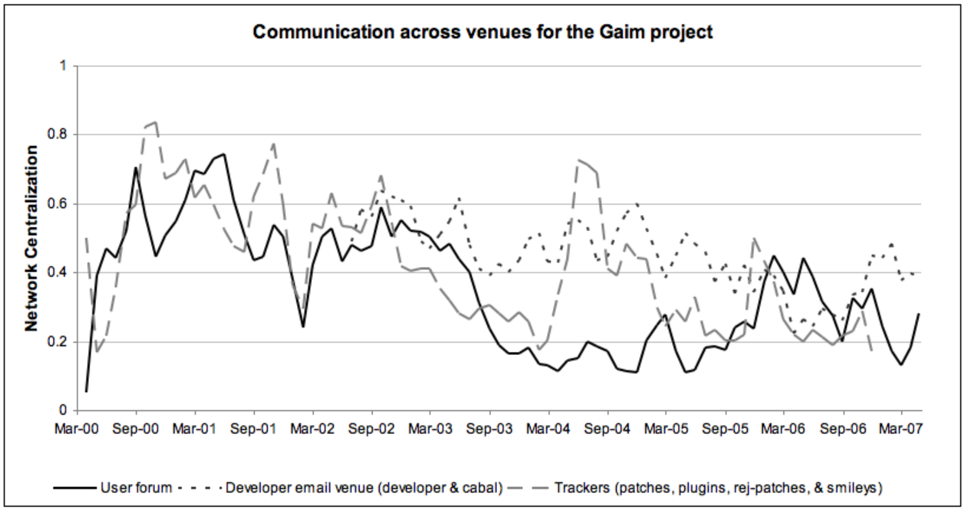

Fig. 4. Dynamics of communication networks in multiple venues for the Gaim project.

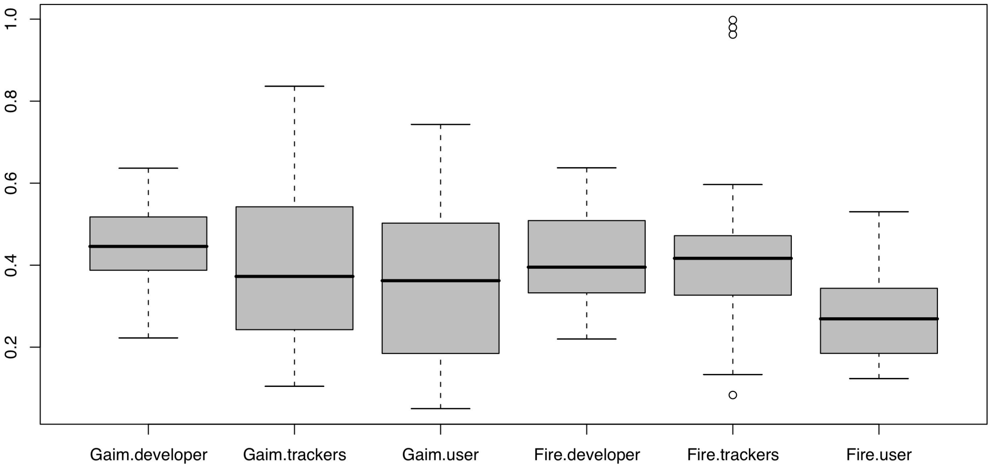

Fig. 5. Boxplots of network centralizations for Fire and Gaim.

has a smaller number of participants. The standard deviations of the centralizations for the user venue and the tracker are comparable, while the standard deviation for the developer list is much lower. Visual inspection of the centralization trends shows that the higher standard deviation reflects a more varied participation dynamic in the user venue and trackers. Periodic spikes in tracker activity appear to indicate project housekeeping, much like the phenomenon observed at the end of the Fire project, as there were several periods during which a large number of tracker items were closed. These large batches of tracker item closings were conducted by one (or very few) individuals, generating the observed highly centralized network structures. While all tracker venues were aggregated for this analysis, the batch tracker item closings were all observed in the bug trackers.

In both Fire and Gaim, the various venues also show a trend toward decreasing centralization of communications, but the end period of the data series appears to show a more stable range of centralization values. In Gaim in particular, this decrease is confirmed by lower standard deviations for the user and tracker venues, both shifting from the series mean, 0.18, to 0.09 during the final two years, suggesting that more stable communication patterns emerged in these venues as the project 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.

Comparison between Fire and Gaim. Our final question concerns the relation between centralization and project effectiveness. While there are obvious limits to the conclusions that can be drawn from just two projects, to our surprise, there were no clear differences in the dynamics of centralization in these two projects. In part this lack of difference reflects the fact that for much of their history, the two projects were in fact comparable in effectiveness. One small difference is in the centralization of communications on the trackers. In Fire, the distribution for the trackers resembles the developer venues more than the user venues, while Gaim shows the opposite pattern. This result may indicate a broader pattern of participation in the Gaim trackers or tighter control in the Fire project. Another exception is the batch closing of tracker requests that were observed in both projects. In Fire, the batch closing happened near the end of the project, apparently as part of closing down the project, while in Gaim they were repeated with some regularity, suggesting regular and ongoing project maintenance tasks that may be relevant for developing a better understanding of project management activities.

VI. Discussion

In this section, we discuss the relevance of our findings related to the methodological issues and research questions introduced above. First, methodologically, we note the importance of applying a dynamic technique to examine the projects. Besides the potential concern for validity based upon communication venue selection for analysis, we note that our analysis demonstrates that centralization in these communication networks varies widely. These data fall into the unstable-dynamic quadrant of Table I, and as such indicate that any construct about which we hope to draw conclusions based upon this data should be expected to vary over time. Collapsing multiple years' worth of interactions to create a single social network would have presented a confused picture and would not have allowed for detailed comparison of the projects.

Turning to our substantive research questions, analysis of the dynamics across venues suggests 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. This further suggests that evidence of leadership activities in the form of public participation will identify different individual contributors as key leaders in each communication venue. This explanation is readily verifiable for the bug tracker venues in which the actions of a single individual make a significant momentary impact on the centralization of the network

Another common feature for all venues in both projects was the overall trend toward decentralization over time. While in both cases, this decentralization reflects a diminished role for the core, in the Fire project, this reduction seems to be the result of the loss of project leadership, while in the effective Gaim project it appears to reflect growth in user participation, complementing the contributions of leaders. Further examination of mailing list archives for each project confirmed that participation trailed off towards the end of the study period in Fire and the concurrent decrease in communication centralization suggests that the most active members of the community had moved on. By contrast, a negative correlation between mailing list participation levels and centralization in Gaim supports the explanation that increased participation, both in number of messages exchanged and number of participants, is the mechanism underlying the trend toward decentralization.

Our conclusion from this comparison is that while the strong contributions of group leaders is apparently important, these are not simply reflected in higher centralization as originally hypothesized. Instead, for FLOSS projects, contributions from the active user base are also important, and these contributions will tend to reduce centralization, as reflected in the decrease in centralization over time. Indeed, there are many potential causes of variation in network structure, as indicated by a shift in the degree of centralization of interactions. The SNA measures cannot tell us why these changes occurred—only that a change has occurred.

In this respect, SNA offers more utility as an indicator of change than for meaningful interpretation. As such, SNA tools may prove useful as a means of filtering large data sets, helping researchers and FLOSS developers alike in narrowing the search space for targeting specific phenomena. Identifying the periods of time during which "something interesting" may have occurred in a stream of 100,000 individual events in a FLOSS project archive may provide useful guidance for more intensive in-depth data analysis. For example, we note an interesting phenomenon that was observed in each of the projects trackers: 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 management practice for a long-term FLOSS project signaling rhythms in group work, believed to be important to success in distributed environments [65]. These spikes in communication centralization in the trackers seems to directly support a form of distributed leadership, with individuals contributing differentially according to the types of tasks managed in each communication venue. Some group members therefore show leadership in managing bugs, while others may show leadership in answering questions from users. Other periods of change in centralization may indicate periods of change in leadership—centralization dips as an old leader steps out, and rises again as a new one takes over. Extending this analysis with a larger number of projects would determine whether this phenomenon is an isolated instance or a more general pattern.

VII. Conclusion

In conclusion, our dynamic analysis of FLOSS group 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 providing evidence that leadership may also manifest differently based on the communicative context. Researchers should therefore be cautious in using single venues to characterize FLOSS 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 of 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, as we argue that this pattern indicates a change in the nature of group leadership.
- The more effective project in our comparison showed greater participation of users in the tracker venues and more periodic housekeeping activities.

Our current study does have some limitations. One concern is about the completeness of our network data. We based our analysis on a variety of developer communications venues. However, we note that FLOSS developers are widely recognized as users of their own products, so we might expect that as developers of IM clients conduct a portion of their communications via IM. Unfortunately, these interactions are not archived and so not available for analysis. The observed communication patterns in these projects may therefore differ from the trends that might be seen in projects that are not developing IM clients, simply due to the nature of the product that they develop. Analysis to explore this possibility remains for future work.

Future research on FLOSS group dynamics could explore the use of SNA to monitor changes in network structures for providing an indicator or diagnostic to help identify potential events of interest for further investigation. This approach may prove useful for filtering other dynamic network data sets to identify events that cause shifts in network structures when the data are too large to be amenable to more direct methods of filtering. The effort required to capture and structure CMC archives to allow this type of analysis is nontrivial, however, so in most cases there is limited practicality in using a data filtering approach based on network dynamics. Finally, future research employing dynamic multi-channel analysis of communication in FLOSS projects at an individual level is needed to establish the degree to which the structure of the entire network reflects distributed forms of project leadership at a more granular level.

Acknowledgment

This research was partially supported by US NSF Grant 05–27457.

References

[1] M. O'Leary and J. Cummings, "The spatial, temporal, and configurational characteristics of geographic dispersion in teams," in *Academy of Management Conference*, Denver, CO, 2002.

[2] M. K. Ahuja, K. Carley, and D. F. Galletta, "Individual performance in distributed design groups: An empirical study," in *Proceedings of the ACM SIGMIS Conference on Computer Personnel Research (CPR)*. San Francisco, CA: ACM, 1997, pp. 160–170.

[3] L. L. Martins, L. L. Gilson, and M. T. Maynard, "Virtual teams: What do we know and where do we go from here?" *Journal of Management*, vol. 30, no. 6, pp. 805–835, 2004.

[4] M. B. Watson-Manheim and F. Belanger, "Support for communication-based work processes in virtual work," *e-Service Journal*, vol. 1, no. 3, pp. 61–82, 2002.

[5] P. C. van Fenema, "Coordination and control of globally distributed software projects," Doctoral Dissertation, Erasmus University, 2002.

[6] P. S. de Souza, "Asynchronous organizations for multi-algorithm problems," Doctoral Thesis, Carnegie-Mellon University, 1993.

[7] D. J. Armstrong and P. Cole, "Managing distance and differences in geographically distributed work groups," in *Distributed Work*, P. Hinds and S. Kiesler, Eds. Cambridge, MA: MIT Press, 2002, pp. 167–186.

[8] W. Scacchi, "The software infrastructure for a distributed software factory," *Software Engineering Journal*, vol. 6, no. 5, pp. 355–369, 1991.

[9] B. Curtis, H. Krasner, and N. Iscoe, "A field study of the software design process for large systems," *Communications of the ACM*, vol. 31, no. 11, pp. 1268–1287, 1988.

[10] D. B. Walz, J. J. Elam, and B. Curtis, "Inside a software design team: knowledge acquisition, sharing, and integration," *Communications of the ACM*, vol. 36, no. 10, pp. 63–77, 1993.

[11] F. P. Brooks, Jr., *The Mythical Man-month: Essays on Software Engineering*. Reading, MA: Addison-Wesley, 1975.

[12] A. Mockus, R. T. Fielding, and J. D. Herbsleb, "A case study of open source software development: The Apache server," in *Proceedings of the International Conference on Software Engineering (ICSE)*, 2000, p. 11 pages.

[13] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE Software*, vol. 18, no. 2, pp. 16–20, March/April 2001.

[14] M. Conway, "How do committees invent," *Datamation*, vol. 14, no. 4, pp. 28–31, 1968.

[15] J. D. Herbsleb and R. E. Grinter, "Architectures, coordination, and distance: Conway's law and beyond," *IEEE Software*, no. September/October, pp. 63–70, 1999.

[16] B. Butler, L. S. Sproull, S. Kiesler, and R. Kraut, “Community effort in online groups: Who does the work and why?” in *Leadership at a Distance*, S. Weisband and L. Atwater, Eds. Mahwah, NJ: Lawrence Erlbaum, 2002.

[17] D. Robey, H. M. Khoo, and C. Powers, “Situated-learning in cross-functional virtual teams,” *IEEE Transactions on Professional Communication*, no. Feb/Mar, pp. 51–66, 2000.

[18] W. Cascio and S. Shurygailo, “E-leadership and virtual teams,” *Organizational Dynamics*, vol. 31, no. 4, pp. 363–376, 2003.

[19] R. K. Hart and P. L. McLeod, “Rethinking team building in geographically dispersed teams: One message at a time,” *Organizational Dynamics*, vol. 31, no. 4, pp. 352–361, 2003.

[20] I. Zigurs, “Leadership in virtual teams: Oxymoron or opportunity?” *Organizational Dynamics*, vol. 31, no. 4, pp. 339–351, 2003.

[21] R. J. House and R. N. Aditya, “The social scientific study of leadership: Quo vadis?” *Journal of Management*, vol. 23, no. 3, pp. 409–473, 1997.

[22] E. A. von Hippel, “Innovation by user communities: Learning from open-source software,” *Sloan Management Review*, vol. 42, no. 4, pp. 82–86, Summer 2001.

[23] E. A. von Hippel and G. von Krogh, “Open source software and the “private-collective” innovation model: Issues for organization science,” *Organization Science*, vol. 14, no. 2, pp. 209–213, 2003.

[24] E. S. Raymond, “Homesteading the noosphere,” *First Monday*, vol. 3, no. 10, 1998.

[25] P. Wayner, *Free For All*. New York: HarperCollins, 2000.

[26] J. L. Berdahl and K. M. Craig, “Equality of participation and influence in groups: The effects of communications medium and sex composition,” *Computer Supported Cooperative Work*, vol. 4, pp. 179–201, 1996.

[27] C. L. Pearce, Y. Yoo, and M. Alavi., “Leadership, social work, and virtual teams: The relative influence of vertical vs. shared leadership in non-profits,” in *Improving Leadership in Non-Profit Organizations*, R. Riggio and S. S. Orr, Eds. San Francisco: Jossey-Bass, 2003.

[28] K. L. Tyran, C. K. Tyran, and M. Shepherd, “Exploring emergent leadership in virtual teams,” in *Virtual Teams That Work: Creating Conditions for Virtual Team Effectiveness*, C. B. Gibbon and S. G. Cohen, Eds. San Francisco: Jossey-Bass, 2003, pp. 183–195.

[29] Y. Yoo and M. Alavi, “Emergent leadership in virtual teams: What do emergent leaders do?” *Information and Organization*, vol. 14, pp. 27–58, 2004.

[30] G. Piccoli, A. Powell, and B. Ives, “Virtual teams: team control structure, work processes, and team effectiveness,” *Information, Technology & People*, vol. 17, no. 4, pp. 359–379, 2004.

[31] N. Misiolek and R. Heckman, “Patterns of emergent leadership in virtual teams,” in *Proceedings of the 38th Hawai’i International Conference on System Sciences*, Big Island, HI, 2005.

[32] S. L. Jarvenpaa, K. Knoll, and D. Leidner, “Is anybody out there? antecedents of trust in global virtual teams,” *Journal of Information Systems*, vol. 14, no. 4, pp. 29–64, 1998.

[33] S. L. Jarvenpaa and D. E. Leidner, “Communication and trust in global virtual teams,” *Organization Science*, vol. 10, no. 6, pp. 791–815, 1999.

[34] F. Sudweeks and S. J. Simoff, “Leading conversations: Communication behaviours of emergent leaders in virtual teams,” in *Proceedings of the 38th Hawai’i International Conference on System Sciences*, Big Island, HI, 2005.

[35] P. Singh, “Open source software development and the small world phenomenon: An empirical investigations of macro level colaboration network properties on project success,” in *Proceedings of the International Conference on Information Systems*, 2007.

[36] S. Christley and G. Madey, “Global and temporal analysis of social positions at SourceForge.net,” in *Proceedings of the Third International Conference on Open Source Systems (OSS)*. Limerick, Ireland: IFIP WG 2.13, 2007.

[37] Y. Gao and G. Madey, “Network analysis of the SourceForge.net community,” in *Proceedings of the Third International Conference on Open Source Systems (OSS), IFIP WG 2.13*, Limerick, Ireland, 2007.

[38] P. V. Singh, Y. Tan, and D. Dey, “Stability and efficiency of communication networks in open source software projects,” in *15th Annual Workshop on Information Technologies & Systems (WITS)*, 2008. [Online]. Available: http://ssrn.com/abstract=886360

[39] S. Christley and G. Madey, “Social positions at sourceforge.net,” Dept. of Computer Science and Engineering, University of Notre Dame,, Tech. Rep., 2007.

[40] L. López-Fernández, G. Robles, J. M. Gonzalez-Barahona, and I. Herraiz, “Applying social network analysis techniques to community-driven libre software projects,” *International Journal of Information Technology and Web Engineering*, vol. 1, no. 3, 2006.

[41] K. Crowston, K. Wei, Q. Li, and J. Howison, “Core and periphery in free/libre and open source software team communications,” in *Proceedings of the 39th Hawai’i International Conference on System System*, Kaua’i, Hawai’i, 2006.

[42] S. Borgatti and M. Everett, “Models of core/periphery structures,” *Social Networks*, vol. 21, pp. 375–395, 1999.

[43] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. New York: Cambridge, 1994.

[44] J. Moreno, “Sociometry and the cultural order,” *Sociometry*, vol. 6, no. 3, pp. 299–344, 1943.

[45] Z. Toeman, “History of the sociometric movement in headlines,” in *Sociometry in France and the United States: A Symposium*. Beacon House, 1950.

[46] L. D. Brandeis, *Other Peoples’ Money, and How the Bankers Use It*. Stokes, 1914.

[47] G. Davis, M. Yoo, and W. Baker, “The small world of the American corporate elite, 1982-2001,” *Strategic Organization*, vol. 1, no. 3, p. 301, 2003.

[48] W. de Nooy, A. Mrvar, and V. Batagelj, *Exploratory social network analysis with Pajek*, ser. Structural analysis in the social sciences. New York: Cambridge University Press, 2005.

[49] R. J. Sandusky and L. Gasser, “Negotiation and the coordination of information and activity in distributed software problem management,” in *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work*. Sanibel Island, Florida, USA: ACM, 2005.

[50] J. Howison, K. Inoue, and K. Crowston, “Social dynamics of FLOSS team communications,” in *Proceedings of the Second International Conference on Open Source Systems*, Como, Italy, 2006.

[51] S. K. Sowe, I. Stamelos, and A. Lefteris, “Identifying knowledge brokers that yield software engineering knowledge,” *Information and Software Technology*, vol. 48, no. 11, 2006.

[52] Y. Long and K. Siau, “Social network structures in open source software development teams,” *Journal of Database Management*, vol. 18, no. 2, pp. 2–40, 2007.

[53] K. Crowston and J. Howison, “Hierarchy and centralization in free and open source software team communications,” *Knowledge, Technology & Policy*, vol. 18, no. 4, pp. 65–85, 2006.

[54] A. Wiggins, J. Howison, and K. Crowston, “Replication of FLOSS research as eResearch,” in *Proceedings of the Oxford e-Research 08 Conference*, Oxford, England, 11–13 September 2008.

[55] M. Everett and S. Borgatti, “Extending centrality,” *Models and Methods in Social Network Analysis*, pp. 57–76, 2005.

[56] L. Freeman, D. Roeder, and R. Mullholland, “Centrality in social networks: II. experimental results,” *Social Networks*, vol. 2, 1980.

[57] D. Braha and Y. Bar-Yam, “From centrality to temporary fame: Dynamic centrality in complex networks,” *Complexity*, vol. 12, 2006.

[58] K. Crowston and J. Howison, “The social structure of free and open source software development,” *First Monday*, vol. 10, no. 2, 2005.

[59] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graphs over time: Densification laws, shrinking diameters and possible explanations,” in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. New York, NY, USA: ACM, 2005, pp. 177–187.

[60] M. Huisman and T. Snijders, “Statistical analysis of longitudinal network data with changing composition,” *Sociological Methods & Research*, vol. 32, no. 2, p. 253, 2003.

[61] J. Howison, M. Conklin, and K. Crowston, “FLOSSmole: A collaborative repository for FLOSS research data and analyses,” *International Journal of Information Technology and Web Engineering*, vol. 1, no. 3, pp. 17–26, 2006.

[62] G. Madey. (2009) SourceForge.net research data.

[63] M. Van Antwerp and G. Madey, “Advances in the SourceForge Research Data Archive (SRDA),” in *Workshop on Public Data about Software Development (WoPDaSD), 4th International Conference on Open Source Systems*, Milan, Italy, September 2008.

[64] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn, “Taverna: A tool for building and running workflows of services,” *Nucleic Acids Research*, vol. 34, no. suppl 2, pp. W729–732, 2006. [Online]. Available: http://nar.oxfordjournals.org/cgi/reprint/34/suppl_2/W729.pdf

[65] M. L. Maznevski and K. M. Chudoba, “Bridging space over time: Global virtual team dynamics and effectiveness,” *Organization Science*, vol. 11, no. 5, pp. 473–492, 2000.