

Understanding Shared Familiarity and Team Performance through Network Analytics

J. Alberto Espinosa
Kogod School of Business
American University
alberto@american.edu

Mark A. Clark
Kogod School of Business
American University
mark.clark@american.edu

Dorothy R. Carter
Department of Psychology
The University of Georgia
dcarter3@uga.edu

Abstract

In this article, we propose a network approach to understanding team knowledge with archival data, offering conceptual and methodological advantages. Often, the degree to which team members' possess shared knowledge has been conceptualized and measured as an aggregate property of a team as a whole. Rather than an aggregate property, however, we argue that shared team knowledge is more appropriately conceptualized as a network of knowledge overlaps or linkages between sets of team members. We created shared knowledge networks for a sample of 1,942 software teams based on members' prior experiences working with one another on different tasks and teams. We included metrics representing topological features of team shared knowledge networks within predictive models of team performance. Our results suggest that network patterning provides additional predictive power for explaining software development team performance over and above the effects of average level of knowledge similarity within a team.

1. Introduction

Teams are fundamental units of work in organizations [1-3] and particularly useful for accomplishing large, complex tasks. To be effective, teams must align with regard to members' "taskwork" (i.e., activities needed to complete the task) and "teamwork" (i.e., activities needed to collaborate). This often requires that members possess somewhat *similar* knowledge related to tasks and one another [4]. When teams come together to perform a collaborative task, they bring with them knowledge derived from their familiarity with prior tasks and teams. Whereas some of this knowledge is held individually, some knowledge is shared among some, or all, other members, forming networked structures of knowledge overlaps among members. We adopt a prior definition of *team knowledge* as "the collection of task- and team-related knowledge held by teammates and their collective understanding of the current situation" [5], encompassing both individually held knowledge as well as the patterns of knowledge overlaps among members [6]. There is increasing evidence that knowledge management (KM)

promotes factors that lead to superior performance [7]. We suggest that understanding the effects of shared knowledge networks on team performance can contribute additional insight for effectively managing knowledge in teams.

Following prior definitions of knowledge workers by Davenport, Reinhardt and colleagues [7, 8], Jennex concluded that engineers are knowledge workers who work on non-routine problems and treat knowledge as their main asset [9]. Software teams in large-scale software production tasks are often composed of software engineers and other knowledge workers who rely on: previously acquired knowledge of tasks (e.g., tools; the product under development; programming languages); and/or one another to do their job. Moreover, research shows knowledge management success is affected by team members' abilities to get the right knowledge to the right user and apply new knowledge to improve performance [10]. A similar argument appears in the literature on expertise coordination [11, 12].

When members' are familiar with similar knowledge sets, they are better prepared to access the right knowledge by the right user at the right time, thus "gluing" individually held knowledge into a coherent whole. Possessing similar knowledge is particularly important when tasks are highly interdependent and intensive activity coordination is critical [13]. Indeed, team knowledge is increasingly recognized as important for organizations in various activities, including research projects and patents [14], collective behavior [15], and coordination across systems [16].

Often, shared knowledge in teams has been conceptualized and measured as a shared or aggregate property of a team as a whole [17]. However, given that different team members may have experienced a variety of different task and team contexts, teams are often characterized by complex patterns of knowledge similarities among members. Thus, adopting a network approach to understanding shared knowledge in teams [18] is likely to be more appropriate for modeling shared knowledge in software teams as compared to an aggregate approach—especially for those teams of mid-to large-size. A network approach can provide more nuanced depictions of the ways in which knowledge is organized and structured in a team as compared to simple aggregate measures. For example, there may be

members who are knowledge isolates, posing potential problems for team collaboration; knowledge similarities may be highly concentrated among a subset of members or widely spread throughout the team; there may be team members that possess similarities with members of disconnected subgroups who share no similarities with one another. Finally, the knowledge that a member shares with two others may not be shared by these other members themselves.

However, except for very few studies, a network approach to knowledge similarity has not been adequately investigated in the extant literature. Much still remains to be understood regarding whether and how knowledge similarities aid team performance. Thus, in this study, we pursue the following research question:

Do properties of team knowledge similarity networks provide additional predictive power to explain software team performance beyond the effects of aggregate measures?

One concern raised with team cognition research is that there is an abundance of constructs defined (e.g., shared knowledge, team mental models, transactive memory, and familiarity, etc.) [19]. In the present study, we focus on shared task and shared team knowledge familiarity linkages as mechanisms for channeling knowledge to the right person for the right task [20]. Familiarity metrics have been used in research for many years [2, 21-25] and can be easily extracted from archival software data. Moreover, researchers on software engineering have developed KM tools to identify and quantify expertise based on such familiarity [26]. In this study, we employ similar familiarity metrics but adapt them to identify overlaps in familiarity among members based on prior experiences with certain tasks/teams.

In the remaining sections of this paper, we present our theoretical arguments and rationale for the importance of the network approach to the study of team knowledge. We then provide our theoretical development and hypotheses related to shared task and team familiarity. We then discuss our methods, present our results, and discuss the implications of our study.

2. Team Knowledge and Network Structure

Knowledge management has been conceptualized in the extant literature as the practice of selectively applying knowledge from previous experiences of decision making to current and future decision making activities with the express purpose of improving the organization's effectiveness, which is enhanced when there are multiple channels for knowledge exchange [20]. Consistent with this view, we argue that knowledge is channeled most effectively towards collective task goals when there is a shared knowledge base that connects the various parts of individual knowledge into

a cohesive whole. This argument parallels those that have been made for various shared cognition constructs like team mental models [27], shared schemas [28] and transactive memory systems [29, 30].

So, what is the best way to represent the shared knowledge of a team? For dyads or 3-member teams with evenly shared knowledge, this question can be readily answered by applying one of the many shared mental model measures in the extant literature. However, with larger teams with knowledge sets that are unevenly shared among members, the representation of team knowledge constructs becomes more complex and simple averages can only provide an incomplete picture. Thus, a network analytic approach, which has been used successfully to study other social structures, is more appropriate for modeling knowledge networks.

Specifically, team knowledge similarity is inherently a social construct leading to other social behaviors – individuals share and exchange knowledge through communication and actions, creating cognitive relationships that help explain team dynamics, process, coordination and performance. A network approach allows the use of methods and tools to analyze complex relations in systems of social actors [31] with knowledge ties between among them [32, 33], describing how team and task familiarity are held, shared, organized and distributed among members, similar to how individual knowledge structure is represented in the cognitive sciences [28, 34, 35].

One strength of the network approach is that team knowledge distribution can be analyzed at any level – individual, dyad, subgroup, ego network, or team – helping to identify important characteristics like centralities (e.g., proportion of knowledge ties to other members), isolates (e.g., members with no knowledge ties to other members) and triads (i.e., transitivity). Such an approach can better explain a team's ability to carry out tasks in ways aggregate team knowledge measures cannot [36]. For example, we can identify knowledgeable members who serve as knowledge exchange hubs, influencing how members coordinate information [37], provide useful knowledge to peers [38], and attain higher individual [39] and team [40] performance. Understanding this is especially useful as teams become larger, with more complex knowledge distributions [16]. Furthermore, popular theories applied successfully to social networks, such as “structural holes” (the lack of links between adjacent individuals, [41] and “weak ties” (connections to those outside one's closest members), [42] can be investigated with team knowledge networks.

Prior seminal research has also employed individual and relational attributes to describe aspects of teamwork and interaction dynamics [43, 44]. Relational knowledge such as shared knowledge, influences how members interact, exchange knowledge [45], communicate [46],

coordinate [47, 48], and perform [49, 50]. These knowledge relationships create complex team knowledge structures that cannot be explained with simple aggregation of the members' knowledge [34], thus the importance of a network approach.

3. Shared Task and Team Familiarity

Jennex and colleagues surveyed 30 members of a knowledge management (KM) journal editorial board and outlined several ways to define KM success, including process success (e.g., productivity). They concluded that KM is successful when knowledge is reused to improve organizational effectiveness by providing the appropriate knowledge to those in need, when they needed it [51]. We suggest that it is the similarities in members' knowledge with regard to "taskwork" and "teamwork" [19], which provides the conduit to locate expertise when needed. Without this overlap, individual knowledge within the team would be disconnected and less effective.

To understand the degree to which members' possess similar knowledge, we leverage the concept of *familiarity*, which in organizational teams, has been defined as "the knowledge that members of a team have about the unique aspects of their work" [52], such as knowledge about the task itself and about other members on the team [53]. As members of a team work together over time, they become familiar with the task domain and with each other [54] and develop a common knowledge base through which team interaction and location of expert sources in the team is facilitated [55]. Thus, members' experiences on prior teams are one way of capturing their level of familiarity with regard to task and/or teamwork.

Studies have shown the positive benefits of familiarity on team performance in mining [22], flight simulation [56], problem solving [24, 53], product development [57], surgical teams [58], and other tasks [2]. For example, a prior study [21] found empirical evidence that individual task and team familiarity increased performance in software teams. Consistent with this work, another study found that shared task knowledge based on peer-rated domain familiarity had a positive effect on team performance when measured as an aggregate - the average task knowledge shared by team members [17]. However, the effect of aggregate shared knowledge disappeared when the analysis included network variables that accounted for more predictive power, which also provided insights into various aspects of performance. For example, the number of isolates had a negative effect on task coordination, whereas task knowledge centralization had a positive effect on strategy coordination, and the proportion of cliques in the team was negatively associated with team cohesion. This prior study provided

empirical evidence that a network analysis perspective can provide more nuanced explanations than individual or aggregated team knowledge measures alone.

In the present study we use archival data to evaluate whether adding team knowledge network variables based on shared familiarity yields deeper insights than those found with individual [22, 25], aggregated [21, 24] familiarity measures on whether such familiarity overlap drives team performance and how. Prior studies have found that effective knowledge management has a positive influence on business process, including productivity [59]. But we argue in this study that it is the knowledge relationship structure within a team which will affect how it interacts and performs. Therefore, we examine whether the structure of shared instances of working together on previous project tasks and with each other are related to team performance. Following this reasoning, we investigate the effects of shared task and team familiarity relational network properties, such as centralization, isolation and transitivity on team performance. We discuss these further next and illustrate them in Figure 1.

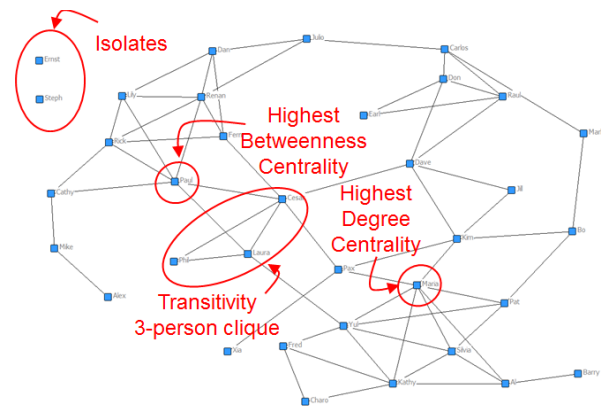


Figure 1. Network variables illustration

Shared Team and Task Familiarity. Because members who have worked together on previous tasks are likely to have had similar learning experiences, we posit that members' shared experiences on prior tasks and/or teams can enhance team performance. Two developers who have worked on the same files and modules in the past are more likely to have better grounding and more shared vocabularies about technical terms and the software product they are working on. Likewise, those who have worked together in the past can be expected to have developed communication patterns and other coordination methods, which may be beneficial to performance. Both types of shared familiarity are likely to help the team develop mutual knowledge [60] and shared mental models [35], which have been found to affect team performance. Therefore, we posit:

H1. Shared task familiarity is positively related to team performance.

H2. Shared team familiarity is positively related to team performance.

Familiarity Centralization. Whereas we argue that shared task and team familiarity are beneficial, we also anticipate that their effects may be influenced by whether familiarity is shared widely across the team or concentrated within a few dyads. Prior studies have found that when team knowledge is centralized, performance diminishes [17, 61]. A centralized network is one in which one or just a few members share a lot of familiarity, whereas the rest not so much. Consequently, we expect a that more widely distributed shared task and team familiarity will be associated with higher levels of performance, compared to networks in which only a few members share familiarity. Therefore, we posit:

H3. Shared task familiarity centralization is negatively related to team performance.

H4. Shared team familiarity centralization is negatively related to team performance.

Familiarity Isolation. Prior studies have also found that knowledge isolates in student teams have a detrimental effect on team outcomes [17]. A member who is not connected to anyone else in the team in terms of shared task and team familiarity has little in common with the rest of the team. This could be beneficial if the isolated member is a very specialized worker, but most often it means that the teammate is just new at the job or has not spent sufficient time developing familiarity with modules, files and teammates. When members have different experiences, lack instances of working together, or are otherwise unfamiliar with one another, we expect to find a negative relationship with multiple performance outcomes relating to process and results. Therefore, we posit:

H5. Shared task familiarity isolation is negatively related to team performance

H6. Shared team familiarity isolation is negatively related to team performance

Transitivity. The effect of transitivity has been studied in depth in the social networks literature. If member A is connected to B and C, these relationships are transitive if B is also connected to C. Consequently, networks with high degree of transitivity are full of “triads”, i.e., 3-member clique triangles. In contrast, links in low transitivity networks exhibit lots of “stars” – i.e. members linked to members who are not connected to each other. The research literature on social network transitivity suggests that transitivity leads to higher integration of members within the network [62]. To the best of our knowledge, there are no studies that have explored the effect of knowledge transitivity in teams. We extrapolate these arguments to team knowledge

networks and argue that, when team members work together in an interdependent task, coordination is enhanced when there is consistency and more integration in the knowledge content being shared. In the context of shared familiarity, three-member cliques will exhibit more homogeneously shared knowledge beyond what single dyads share, which will help members anticipate each other’s actions and perform better. Thus, we posit:

H7. Shared task familiarity transitivity is positively related to team performance.

H8. Shared team familiarity transitivity is positively related to team performance.

4. Network Analytics Approach

We employ a network analytic approach using software production data from an archival data set and capture team knowledge structure metrics represented by shared familiarity relational ties [63]. We describe how we computed each network variable in the next section. The overall strategy was to extract software production data from a configuration management system and then use this data to construct network edgelist tables that contained every pair of developers in each team, with archival data statistics about their shared familiarity with same software modules and same files, and also with familiarity working with each other over time. We then used these shared familiarity metrics as tie weights, which were then used in the computation of network variables. Using archival data to compute these network structure variables is a powerful way to analyze teams without the bias of self-report surveys. This approach allows for the computation of aggregate measures, while retaining the structural detail that comprises knowledge in the team, which is particularly useful for larger teams.

5. Sample and Data Analysis

We employed archival data from software development teams for this study. We chose the software task because its activities are highly interdependent and therefore require a substantial amount of knowledge sharing to carry out the task effectively. Software tasks are very useful for archival data analysis studies because most software organizations keep detailed software production records and statistics. Most large software development organizations employ configuration or management systems (CMS) to help developers coordinate software modifications with others. For example, when a developer needs to make a change on a given file (i.e., called a “delta), the CMS locks any file that may be dependent on the file being modified until the work with that file is complete. When the delta is finished, the file is checked back in and the CMS unlocks the dependent files. More importantly for our study, the

CMS is constantly recording statistics about each modification, such as: who made the modification and when; how long it took to complete; how many lines of code were added, deleted or modified; and which modules, files and subsystems were affected, among many other things. While such datasets do not contain validated scales or attitudinal variables, the records available provide an excellent source of data to study software team performance by constructing individual, dyadic and team level variables.

Software modifications in this organization are done via “modification requests” (MRs) and “deltas”. An MR is an approved project to either develop new functionality or update existing functionality. An MR in the organization we studied is implemented by a dedicated team, which generally stays together until the completion of the MR. An MR has a formally approved budget, team members and other resources. Naturally, an MR generally contains more than one software change. A “delta” is the basic unit of software change in this organization, consisting of the changes made to a single software file by one developer during a single check-out/check-in cycle. Deltas are carried out by members of the MR team, which allowed us to compute knowledge relationships between all pairs of developers in each MR team, based on their familiarity with each other and with specific software modules and files.

We analyzed data from 1,942 MRs containing a total of 672,209 deltas. This is a large software product and it contains several thousand MRs. For this study, we used all MR’s for teams of 6 to 12 members. We selected teams of 6 or more members because some network structural properties are not as observable or relevant with smaller teams. We limited the team size to 12 because as teams get larger some of the network metrics (e.g., network density) become more sparse. Generally, larger networks are typically analyzed individually, which is what we are doing in a separate study. In the present study, we compute network metrics for each of the MR teams and then use these metrics as variables in a predictive regression model.

Performance in software development is generally measured in terms of process (e.g., on time, on budget) and product performance (e.g. meet requirements, few errors) [64-66]. To test our hypotheses we constructed a regression model with software development time (in days) as the outcome variable, along with various predictors. We selected software development time (reversed) as our dependent variable because it is one of the most widely used measures of software process performance [67, 68]. We tested the percentage of error repair deltas in MRs as a measure of product performance, but we did not find interesting results, most likely because MR’s must be error free after testing, so more error repairs simply increase the development time.

Because QQ-Plots revealed that the regression residuals were not normal, we log-transformed this variable, yielding normally distributed residuals. Also, because we were interested in a performance metric, we modeled the negative value of the logged variable (i.e., higher values represent less development time, which is considered higher performance).

We started with a baseline regression model with several control variables known to affect software development time, including: team size; number of modules spanned by the MR (i.e., a proxy for software complexity); number of sub-systems affected by the modification (i.e., another measure of software complexity); and effort distribution (i.e., dispersion in the number of deltas contributed by each team members using a Gini coefficient of homogeneity ranging from 0 if one person developed all deltas, to 1 if each developer contributed an equal number of deltas). We then specified a full regression model by adding to the baseline model the network structure variables we constructed. We evaluated all possible regression models between the baseline and full model using the Stepwise variable selection method. The criteria for removal or inclusion of variables in the Stepwise procedure was set at $p=0.15$, which is customary in the Stepwise method, which is a more inclusive threshold.

To construct the network variables we adapted measures from previous studies of familiarity, which are generally based on counts of how many times an individual has worked in the past on a given task (i.e., task familiarity) [21, 23, 25] and with other members [21, 24] (i.e., team familiarity). When looking at a given measure of familiarity for any two members, some of that familiarity will be unique to each individual (i.e., unshared) and some will be overlapping (i.e., shared). To construct the shared familiarity version of these metrics we used similar counts, but only counting the modules, files and projects in which both members of the dyad had worked. We used a predictive analytics method, blending standard network analysis to compute network structure variables for each of the teams in our sample, and traditional regression methods to develop the actual predictive model. For each MR team, we analyzed each dyad in the MR team and then computed:

- **Shared Task Familiarity (Modules)** as the total number of modules in which both members of the dyad had contributed deltas to in the past.
- **Shared Task Familiarity (Files)**, same as (a) but using the total number of files in which both members had contributed deltas to in the past.
- **Shared Team Familiarity (MR’s)** as the number of times the pair had worked in the same MRs in the past.

We then wrote an R script to loop through all the dyadic relationships for each MR to construct the following network structure variables for each team, and

for each of the three shared familiarity metrics (i.e., team, task-modules and task-files):

- **Degree Centralization** – degree centrality is the number of ties a member has with other members (i.e., how many modules, files and projects he/she shares with others). Degree centralization measures how much members’ centralities deviate from the most central member, with higher values indicating more centralized shared knowledge.
- **Betweenness Centralization** – betweenness centrality for a member measures how many pairs have to go through that member to connect with each other. Betweenness centralization measures how much members’ centrality deviate from the most central member. The higher the number, the more the team relies on one or few members to connect with each other.
- **Isolation** – the proportion of members who have no shared familiarity connection with others in the team.
- **Transitivity** – the proportion of triangle connections, relative to all possible triangles for a team of that size.

6. Results

Table 1 shows the summary results for the best regression model produced by the Stepwise method. As we discussed earlier, we are only showing the best set of predictors identified by the Stepwise process. This table only shows predictors that remained significant at the $p < 0.15$ level as variables were removed and added to the model. We focus our discussion of results on the most significant predictors at $p < 0.05$, with a brief mention to the remaining predictors. In reference to the control variables, all four variables were significant, as expected. Controlling for other variables, team performance diminished with team size and number of sub-systems. This result shows that software development is faster when teams are small and the software is less complex. Interviews conducted in prior studies [citation omitted for anonymity] revealed that MR teams work better when they were small because of the amount of coordination required in this type of work. Interestingly, team performance increased with the number of modules. This result seems counter intuitive, but we attribute this to the slight correlation in the number of sub-systems and modules; once we controlled for the number of subsystems, MRs that span more modules were completed faster. Performance also improved when effort distribution was more uneven, which was expected. When most of an MR development effort is carried out by a few software developers in the team, it took less time to complete than when development was more evenly spread across many developers.

Consistent with H1 and H2, average shared task and team familiarity were strong positive predictors of team performance ($\beta = 50.374$, $p < 0.001$ for average shared

team familiarity (MR’s); and $\beta = 7.92$, $p < 0.001$ for average shared task familiarity (Modules). However, when the remaining network structural variables were added to the model, their effect changed substantially and the predictive power of the model increased significantly ($p < 0.001$). As Table 1 shows, the effect of average shared task familiarity (Modules) became negative and significant, and the effect for shared team familiarity remained positive but its p-value became marginally non-significant. These results provide strong support for our argument that network structural variables not only have strong predictive power over team performance, but can change results previously believed to be true, while providing a more nuanced understanding of the drivers of team performance.

Table 1. Regression results

Regression Model on Software Team Performance			
Variable	Baseline Model		
	Coefficient	Std. Error	P-Value
Team Size	-0.102	0.021	<0.001
Number of Modules	0.002	0.001	0.001
Number of Sub-Systems	-0.061	0.001	<0.001
Effort Distribution (evenness)	2.892	0.324	<0.001
Average Shared Task Familiarity (Modules)	-0.014	0.005	0.003
Average Shared Team Familiarity (MRs)	0.049	0.030	0.104
Shared Team Familiarity Distribution (MR’s)	-0.023	0.011	0.034
Betweenness Centralization (Modules)	0.490	0.198	0.013
Degree Centralization (Files)	-0.780	0.221	<0.001
Betweenness Centralization (Files)	0.985	0.225	<0.001
Degree Centralization (MRs)	-0.295	0.148	0.046
% Isolates (Modules)	-0.674	0.120	<0.001
% Isolates (MRs)	-0.399	0.197	0.043
Transitivity (Files)	0.000	0.000	0.143
Transitivity (MRs)	0.001	0.001	0.085
N			1,926
Adjusted R2			0.205
R2			0.211
p-value			<0.001

Consistent with H3, shared team familiarity distribution had a negative effect on team performance, suggesting that having more widely shared team familiarity helps performance. Also, consistent to H4, we found that degree centralization of shared task familiarity (Files) had a negative effect on team performance, suggesting that more widely distributed shared task familiarity improves team performance. Surprisingly, the effect of betweenness centralization of shared task familiarity (Modules) had a positive effect on team performance. This result is the perfect illustration of why the network perspective matters when studying team performance. While one measure of centralization (degree) had one effect, the other (betweenness) had the opposite effect. Consistent with arguments in weak ties research [42], densely connected network actors may get overwhelmed. In contrast, consistent with “structural holes” arguments [69], a high degree of betweenness centralization can help members find and access specialized knowledge when needed.

Consistent with H5 and H6, shared task and team familiarity isolation was detrimental to performance. Having members with disconnected task and team familiarity from the rest of the team is never a good thing. Finally, we did not find support for H7 and H8 about transitivity, but the positive and marginally non-significant effects from these variables point to an effect in this direction, albeit these effects are not significant at the $p < 0.05$ level. It may be possible that with a larger data set, these effects will become significant.

7. Discussion

The value of the network perspective resides not only on the ability to represent heterogeneous team knowledge structures better than with methods suited to homogeneous knowledge [see 5, 70 for further methods], but also for its facility in modeling dyadic knowledge relationships across multiple content areas in any way suitable to a particular research inquiry. For example, a dyad relationship could be modeled as knowledge similarity (e.g., shared mental model) or as distance or difference (e.g., knowledge disconnect). Further, when all the dyadic knowledge relationships are modeled into a single team knowledge network, this approach can be used to derive metrics of heterogeneity using popular measures like Gini coefficients [71]. It also facilitates the identification of important sub-groups, such as factions, communities and clusters.

Our illustration shows that team network structural properties help us enhance our understanding of how knowledge operates within a team to influence performance. While shared knowledge has been represented through a variety of measures and methods in the past – e.g., task relatedness matrices [34], quadratic assignment procedure [35] and schema agreement [28] – our approach allows the incorporation of any of these measures into a team knowledge network that includes all dyads, allowing not only for richer analysis and more nuanced explanations of team processes and outcomes, but also for aggregation into more general measures.

Without a network perspective, aggregate measures provide an incomplete or incorrect picture of a team's knowledge structure, especially for larger teams. Our study shows that adding network structure variables to predictive models of team performance not only provides a more nuanced understanding of how various aspects of team knowledge affect different aspects of performance, but may also change the direction and significance of prior effects. In our model, an ANOVA test between the model with only aggregate shared familiarity variables and the final model with significant variables, showed a significant increase in predictive power ($F=19.981, p < 0.001$).

8. Conclusions

While these results are preliminary, they underscore the importance of incorporating a network analysis perspective into research models of team performance. Furthermore, our study contributes methodologically by providing a way to develop team knowledge networks from archival data. Researchers have been studying teams, team cognition and social networks for decades. While much progress has been made during this time, much remains to be learned. The team cognition literature is replete with constructs, measures and methods to study team knowledge. Some of these methods are very sophisticated and effective at measuring knowledge constructs. However, most of these metrics apply to either dyads or very small teams. We argue that we can use any dyadic measure of team knowledge and measure it for each dyad in the team and compose a knowledge network. In our study we have developed shared familiarity networks, which to the best of our knowledge, has not been done before.

Similarly, social network analysis theories and methods have been around since the 1930's and perhaps earlier. But these methods and theories have generally been applied to behavioral aspects of social interaction (e.g., friendship, communication, advice, supervision). Network analytics methods have also been applied to many more fields, including physics, biology, electronics and computing, among many others. It has had only limited application to the study of team networks, which is distinct from the use of network analysis methods to measure team knowledge constructs.

For example, the "task-relatedness" matrix is a very popular method used to measure shared mental models. Each member fills in a matrix to rate which task components are related to which task components. The correlation or closeness of the matrices of two individuals is typically analyzed with network analysis methods to impute a metric of knowledge structure similarity and labeled a "shared mental model". However, this metric is for dyads, not for entire teams. We argue that if we do this for every dyad in a team we can compose a full team knowledge network. To the best of our knowledge, this latter aspect has not been pursued in depth in the extant research literature. This perspective offers endless possibilities to analyze how teams organize their knowledge, tasks and communication to work together. Furthermore, many other non-cognitive networks can also be developed to measure things like geographic dispersion, time zone differences, communication volumes and friendship, among many others, to build more complete predictive models of team performance. Our study is a step in that direction.

Our study also contributes to practice. It would be a straightforward exercise to develop visual network diagrams of the shared task and team familiarity

networks of software developers in a firm and help management form teams with a desired shared familiarity structure. Researchers have developed similar tools to identify individual familiarity with software modules and files, thus helping managers and developers locate expertise [26].

9. Limitations

Our study may have a number of limitations, with three that are important to note. First, the analysis is based on archival data, which has inherent limitations. For example, this archival data does not have explicit data about team process and knowledge. However, we have illustrated a novel approach to extract useful data to create related proxy measures. For example, prior seminal studies of familiarity [22, 23, 52] have relied on reductions in task completion times to evaluate task and team familiarity. Our approach employs more sophisticated methods to test similar outcomes, examining shared familiarity in software teams.

The second major limitation is that team members work sometimes with the same peers over time. Therefore the OLS assumption of independence may not hold up. We have conducted some tests of independence with encouraging results and have developed models using traditional time series corrections like lagging variables. Our results are somewhat similar, but we need further analysis to determine this, which we plan to undertake in a follow up study.

Finally, most of the network variables we used are based on binary network constructs. For example, degree centrality counts a tie when it exists, regardless of the value of the tie. So, two members that have worked together on just one project will have a tie, just as two members who have worked together many times. In our follow up study we will integrate tie weights into our analysis. We anticipate stronger results once tie weights are factored in.

Despite these limitations, our study makes important contributions to team knowledge research, including that it: leverages the power of network theories and methods; builds upon strengths of current team cognition measures; is computationally simple; can be used at both, aggregate and detail levels; incorporates both, individual and relational knowledge attributes, providing a complete picture of the team's knowledge; allows for the computation and visual representation of various team's knowledge measures; and provides a richer explanation of the effect of structural aspects of team knowledge on team outcomes. Our provides evidence that this network perspective adds explanatory value. While further development and testing of network methods to study team knowledge are still needed, our study shows promise to inform research and practice how team knowledge can influence performance.

10. References

- [1] S. G. Cohen and D. E. Bailey, "What Makes Teams Work: Group Effectiveness Research From the Shop Floor to the Executive Suite," *Journal of Management*, vol. 23, pp. 239-290, 1997.
- [2] D. A. Harrison, S. Mohammed, J. E. McGrath, A. T. Florey, and S. W. Vanderstoep, "Time Matters in Team Performance: Effects of Member Familiarity, Entrainment, and Task Discontinuity on Speed and Quality," *Personnel Psychology*, vol. 56, pp. 633-669, 2003.
- [3] J. McGrath, "Time, Interaction and Performance (TIP)," *Small Group Research*, vol. 22, pp. 147-174, 1991.
- [4] R. J. Klimoski and S. Mohammed, "Team Mental Model: Construct or Metaphor," *Journal of Management*, vol. 20, pp. 403-437, 1994.
- [5] N. J. Cooke, E. Salas, J. A. Cannon-Bowers, and R. J. Stout, "Measuring Team Knowledge," *Human Factors*, vol. 42, pp. 151-173, 2000.
- [6] A. W. Wooley, C. F. Chabris, A. Pentland, N. Hashmi, and T. W. Malone, "Evidence for a Collective Intelligence Factor in the Performance of Human Groups," *Science*, vol. 330, pp. 686-688, 2010.
- [7] T. H. Davenport, *Thinking for a Living: How to Get Better Performances And Results from Knowledge Workers* Harvard Business Press, 2005.
- [8] W. Reinhardt, B. Schmidt, P. Sloep, and H. Drachsler, "Knowledge Worker Roles and Actions: Results of Two Empirical Studies," *Knowledge and Process Management*, vol. 18, pp. 150-174, 2011.
- [9] M. E. Jennex, "Knowledge Management Success in an Engineering Firm," *Engineering Management Reviews*, vol. 2, pp. 65-74, 2013.
- [10] M. E. Jennex, S. Smolnik, and D. Croasdell, "Where to Look for Knowledge Management Success," in *45th. Hawaii International Conference on System Sciences*, Maui, Hawaii, 2012.
- [11] S. Faraj and L. Sproull, "Coordinating Expertise in Software Development Teams," *Management Science*, vol. 46, pp. 1554-1568, December 2000.
- [12] A. Tiwana and E. R. McLean, "Expertise Integration and Creativity in Information Systems Development," *Journal of Management Information Systems* vol. 22, pp. 13 - 44, Summer 2005.
- [13] T. Malone and K. Crowston, "The Interdisciplinary Study of Coordination," *ACM Computing Surveys*, vol. 26, pp. 87-119, 1994.
- [14] S. Wutchy, B. F. Jones, and B. Uzzi, "The Increasing Dominance of Teams in Production of Knowledge," *Science*, vol. 316, pp. 1036-1039, 2007.
- [15] L. A. DeChurch and J. R. Mesmer-Magnus, "The Cognitive Underpinnings of Effective Teamwork: A Meta-Analysis," *Journal of Applied Psychology*, vol. 95, pp. 32-53, 2010.

- [16] A. Majchrzak, S. L. Jarvenpaa, and A. B. Hollingshead, "Coordinating Expertise Among Emergent Groups Responding to Disasters," *Organization Science*, vol. 18, pp. 147-161, 2007.
- [17] J. A. Espinosa and M. A. Clark, "Team Knowledge Representation: A Network Perspective," *Human Factors*, vol. 56, pp. 333 - 348, March 2014.
- [18] J. N. Mell, D. van Knippenberg, and W. P. van Ginkel, "The Catalyst Effect: The Impact of Transactive Memory System Structure on Team Performance," *Academy of Management Journal*, vol. 57, pp. 1154-1173, 2014.
- [19] J. A. Cannon-Bowers and E. Salas, "Reflections on Shared Cognition," *Journal of Organizational Behavior*, vol. 22, pp. 195-202, 2001.
- [20] M. E. Jennex, S. Smolnik, and D. Croasdell, "Knowledge Management Success in Practice," in *47th. Hawaii International Conference on System Sciences*, Maui, Hawaii, 2014.
- [21] J. A. Espinosa, S. A. Slaughter, R. E. Kraut, and J. D. Herbsleb, "Familiarity, Complexity and Team Performance in Geographically Distributed Software Development," *Organization Science*, vol. 18, pp. 613-630, July-August 2007.
- [22] P. S. Goodman and D. P. Leyden, "Familiarity and Group Productivity," *Journal of Applied Psychology*, vol. 76, pp. 578-586, 1991.
- [23] P. S. Goodman and S. Shah, "Familiarity and Work Group Outcomes," in *Group Processes and Productivity*, S. Worchel, W. Wood, and J. A. Simpson, Eds., ed Newbury Park, CA: Sage Publications, 1992, pp. 578-586.
- [24] D. H. Gruenfeld, E. A. Mannix, K. Y. Williams, and M. A. Neale, "Group Composition and Decision Making: How Member Familiarity and Information Distribution Affect Process and Performance," *Organizational Behavior and Human Decision Processes*, vol. 67, pp. 1-15, 1996.
- [25] R. Reagans, L. Argote, and D. Brooks, "Individual Experience and Experience Working Together: Predicting Learning Rates from Knowing Who Knows What and Knowing How to Work Together," *Management Science*, vol. 51, pp. 869-881, June 2005.
- [26] A. Mockus and J. D. Herbsleb, "Expertise Browser: A Quantitative Approach to Identifying Expertise," in *24th International Conference on Software Engineering*, Orlando, Florida, 2002, pp. 503-512.
- [27] S. Mohammed, L. Ferzandi, and K. Hamilton, "Metaphor no More: A 15-Year Review of the Team Mental Model Construct" *Journal of Management*, vol. 36, 2010.
- [28] J. R. Rentsch and R. J. Klimoski, "Why do Great Minds Think Alike?: Antecedents of Team Member Schema Agreement," *Journal of Organizational Behavior*, vol. 22, pp. 107-120, 2001.
- [29] K. Lewis, D. Lange, and L. Gillis, "Transactive Memory Systems, Learning and Learning Transfer," *Organization Science*, vol. 16, pp. 581-598, 2005.
- [30] I. Oshri, P., v. Fenema, and J. Kotlarsky, "Knowledge Transfer in Globally Distributed Teams: the Role of Transactive Memory," *Information Systems Journal*, vol. 18, pp. 593-616, 2008.
- [31] K. Carley and D. Krackhardt, "Cognitive Inconsistencies and Non-Symmetric Friendship," *Social Networks*, vol. 18, pp. 1-27, 1996.
- [32] J. Scott, *Social Network Analysis*. Thousand Oaks, CA: Sage Publications, 2012.
- [33] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*: Cambridge University Press, 1994.
- [34] N. J. Cooke, P. A. Kiekel, E. Salas, R. J. Stout, C. Bowers, and J. A. Cannon-Bowers, "Measuring Team Knowledge: A Window to the Cognitive Underpinnings of Team Performance," *Group Dynamics: Theory, Research and Practice*, vol. 7, pp. 179-199, 2003.
- [35] J. Mathieu, G. F. Goodwin, T. S. Heffner, E. Salas, and J. A. Cannon-Bowers, "The Influence of Shared Mental Models on Team Process and Performance," *Journal of Applied Psychology*, vol. 85, pp. 273-283, 2000.
- [36] A. Tziner and D. Eden, "Effects of Crew Composition on Crew Performance: Does the Whole Equal the Sum of its Parts?," *Journal of Applied Psychology*, vol. 70, 1985.
- [37] S. L. Jarvenpaa and A. Majchrzak, "Knowledge Collaboration Among Professionals Protecting National Security: Role of Transactive Memories in Ego-Centered Knowledge Networks," *Organization Science*, vol. 19, pp. 377-379, 2009.
- [38] M. Wasko, S. and S. Faraj, "Why Should I Share? Examining Social Capital and Knowledge Contribution in Electronic Networks of Practice," *MIS Quarterly*, vol. 29, 2005.
- [39] M. K. Ahuja, D. G. Galletta, F., and K. Carley, "Individual Centrality and Performance in Virtual R&D Groups: an Empirical Study," *Management Science*, vol. 49, pp. 21-38, 2003.
- [40] S. Huang and J. N. Cummings, "When Critical Knowledge is Most Critical: Centralization in Knowledge-Intensive Teams," *Small Group Research*, vol. 42, pp. 669-699, 2011.
- [41] R. S. Burt, *Structural Holes : The Social Structure of Competition*
Cambridge, Massachusetts: Harvard University Press, 1995.
- [42] M. S. Granovetter, "The Strength of Weak Ties," *American Journal of Sociology*, vol. 78, pp. 1360-1380, May 1973 1973.
- [43] S. Barley, "Technology as an Occasion for Structuring: Evidence from Observations of CT Scanners and the Social Order of Radiology Departments," *Administrative Science Quarterly*, pp. 78-108, 1986.
- [44] S. Barley, "The Alignment of Technology and Structure Through Roles and Networks," *Administrative Science Quarterly*, vol. 35, pp. 61-103, 1990.
- [45] K. Carley, "An Approach for Relating Social Structure to Cognitive Structure," *Journal of Mathematical Sociology*, vol. 12, pp. 137-189, 1986.

- [46] C. D. Cramton, "The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration," *Organization Science*, vol. 12, pp. 346-371, 2001.
- [47] J. A. Cannon-Bowers, E. Salas, and S. Converse, "Shared Mental Models in Expert Team Decision-Making," in *Individual and Group Decision-Making: Current Issues*, J. Castellan, Ed., ed Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., 1993, pp. 221-246.
- [48] R. Rico, M. Sánchez-Manzanares, F. Gil, and C. Gibson, "Team Implicit Coordination Processes: A Team Knowledge-Based Approach," *Academy of Management Review*, vol. 33, pp. 163-184, 2008.
- [49] E. E. Entin and D. Serfaty, "Adaptive Team Coordination," *Journal of Human Factors*, vol. 41, pp. 321-325, 1999.
- [50] R. J. Stout, J. A. Cannon-Bowers, E. Salas, and D. M. Milanovich, "Planning, Shared Mental Models, and Coordinated Performance: An Empirical Link is Established," *Human Factors*, vol. 41, pp. 61-71, 1999.
- [51] M. E. Jennex, S. Smolnik, and D. T. Croasdell, "Towards a Consensus Knowledge Management Success Definition," *VINE Journal of Information and Knowledge Management Systems*, vol. 39, pp. 174-188, 2009.
- [52] P. S. Goodman and S. Garber, "Absenteeism and Accidents in a Dangerous Environment: Empirical Analysis of Underground Coal Mines," *Journal of Applied Psychology*, vol. 73, pp. 81-86, 1988.
- [53] G. Littlepage, W. Robison, and K. Reddington, "Effects of Task Experience and Group Experience on Group Performance, Member Ability, and Recognition of Expertise," *Organizational Behavior and Human Decision Processes*, vol. 69, pp. 133-147, 1997.
- [54] R. Katz, "The effects of group longevity on project communication and performance," *Administrative Science Quarterly*, vol. 27, pp. 81-104, 1982.
- [55] M. Alavi and D. E. Leidner, "Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," *MIS Quarterly*, vol. 25, pp. 107-136, March 2001 2001.
- [56] B. G. Kanki and H. C. Foushee, "Communication as Group Process Mediator of Aircrew Performance," *Aviation, Space, and Environmental Medicine*, vol. 60, pp. 402-410, May 1989.
- [57] M. E. Sosa and F. Marle, "Assembling Creative Teams in New Product Development Using Creative Team Familiarity," *Journal of Mechanical Design*, vol. 135, p. 081009, 2013.
- [58] A. Kurmann, S. Keller, F. Tschan-Semmer, J. Seelandt, N. K. Semmer, D. Candinas, *et al.*, "Impact of Team Familiarity in the Operating Room on Surgical Complications," *World Journal of Surgery*, vol. 38, pp. 3047-3052, 2014.
- [59] M. E. Jennex, S. Smolnik, and D. Croasdell, "The Search for Knowledge Management Success," in *49th. Hawaii International Conference on System Sciences*, Kauai, Hawaii, 2016.
- [60] R. Krauss and S. Fussell, "Mutual Knowledge and Communicative Effectiveness," in *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*, J. Galegher, R. E. Kraut, and C. Egidio, Eds., ed Hillsdale, NJ: Lawrence Erlbaum, 1990, pp. 111-146.
- [61] D. L. Rulke and J. Galaskiewicz, "Distribution of Knowledge, Group Network Structure, and Group Performance," *Management Science*, vol. 46, pp. 612-625, 2000.
- [62] H. Louch, "Personal Network Integration: Transitivity and Homophily in Strong-Tie Relations," *Social Networks*, vol. 22, pp. 45-64, 2000.
- [63] K. Carley, "Extracting Team Mental Models Through Textual Analysis," *Journal of Organizational Behavior*, vol. 18, pp. 533-558, 1997.
- [64] J. G. Coopridge and J. C. Henderson, "Technology-Process Fit: Perspectives on Achieving Prototyping Effectiveness," *Journal of Management Information Systems*, vol. 7, pp. 67-87, 1991.
- [65] S. R. Nidumolu, "The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable," *Information Systems Research*, vol. 6, pp. 191-219, 1995.
- [66] B. H. Wixom and H. J. Watson, "An empirical investigation of the factors affecting data warehousing success," *MIS Quarterly*, vol. 25, pp. 17-41, 2001.
- [67] A. Gopal, J. A. Espinosa, S. Gosain, and D. P. Darcy, "Coordination and Performance in Global Software Service Delivery: The Vendor's Perspective," *IEEE Transactions on Engineering Management* vol. 58, pp. 772-785, 2011.
- [68] J. Herbsleb and D. Moitra, "Global Software Development," *IEEE Software*, vol. 18, pp. 16-20, March/April 2001.
- [69] R. S. Burt, "Structural Holes and Good Ideas," *American Journal of Sociology*, vol. 110, pp. 249-399, 2004.
- [70] K. Lewis, "Measuring Transactive Memory Systems in the Field: Scale Development and Validation," *Journal of Applied Psychology*, vol. 88, pp. 587-604, August 2003.
- [71] R. Dorfman, "A formula for the Gini Coefficient," *The Review of Economics and Statistics*, vol. 61, pp. 146-156, 1979.