

Impact of Socio-Technical Network on Process Performance

Rong Liu

IBM T.J. Watson Research Center
New York, USA
rliu@us.ibm.com

Akhil Kumar

Smeal College of Business, Penn State University
University Park, PA 16802, USA
AkhilKumar@psu.edu

Abstract—There is a growing interest in socio-technical networks that encompass people, process and technology. They capture connections among technical artifacts and human resources. Existing studies have shown that social networks among resources that collaborate to work on a business process instance have an impact on the performance of the instance. Hence, the social network among the resources should be aligned with the technical work structure. Thus, we propose to identify empirically the specific features of social networks (e.g. size, closeness, density, etc.) that have the most significant impact on the performance of a process. Then, we can leverage these significant social network features to optimize dynamic assignment of tasks to resources for better performance. In general, organizations can use this two-part approach to evaluate their resource social networks and shape their resource networks to achieve socio-technical network alignment. We use real data from business processes in the IT incident management domain for this study to demonstrate our proposed methodology.

Keywords– business process, collaborative social network, socio-technical network, social BPM, social network metrics

I. INTRODUCTION

In recent years there has been increasing research interest in combining social and technical networks to improve performance of a team process. A socio-technical network lies at the confluence of people, process and technology. Conway's law [9] proposed a long time ago states that organizations have a tendency to design systems that mirror their communication structures. Other researchers have suggested that software should be conceived of as a socio-technical concept [2]. They argue that design, execution and productivity of software process models can be improved by mapping the social network among developers who contribute to components of a software application onto the network of connections among the software components. It was shown [6] that the failure-proneness of a software component can be predicted more accurately by considering the combined socio-technical network of a software application.

Research in healthcare [15] has shown that communication breakdowns among medical professionals can lead to adverse effects on surgical patients. These breakdowns result from poor handoffs involving verbal communications and ambiguities about responsibilities. Related research in the same stream of work [14] has shown that often the members of a surgical team that performs an operation have never even met each other

leading to adverse outcomes. From these and other studies there is ample evidence to suggest that the nature of the social network among participants who collaborate in a team process does have an impact on the outcome and performance of the process in terms of quality, failure rate and other measures. However, it is not clear what specific social network factors affect key performance metrics in various domains.

Thus, it is important to determine which specific features of the associated social network have the most significant impact on the performance of a business process. This can lead to a deeper understanding of how social networks affect a process performance and also to actionable guidelines for leveraging social networks in practice. First, these features can be utilized to optimize resource allocation. For instance, [6] reported that software failures can be predicted by over a dozen of social-technical networks metrics such as centrality, reachability, and hierarchy. However, the finding would be more useful if the study can pinpoint the most significant features, e.g. reachability or centrality. If, say, reachability is the most significant factor, optimizing it among resources can be a goal when assigning resources to software development tasks.

Second, it is well accepted that the resource social networks need to be in synergy with technical structure of tasks to facilitate resource collaboration. Identifying the most significant features of social networks can help organizations evaluate the fitness of social networks and align them with the technical structure of tasks. Moreover, business processes in different domains have different technical structures and task dependencies. Hence, a finding from one domain may not apply to another, and the analysis has to be done case by case. Often statistical analysis is needed to discover significant features of social networks. Following the finding, rigorous mathematical algorithms can be used to leverage these features to lead to optimal resource allocation and optimize process performance.

In this paper, we wish to test whether certain features or metrics of resources' social networks contribute to overall business process performance using a real example in IT Incident Management domain. Since there are many metrics of a social network, we also desire to understand which if any of them play a larger role. We study real data from incidents and evaluate how throughput times of incidents depend on various metrics such as closeness and density among resources that work on an incident. We would also like to gain a better

understanding of the varying nature and intensity of interactions among teams since any business process is a team process involving the various resources that perform its different steps. Considering a process involving n resources, at the "low-interaction end" where each interaction occurs sequentially, there are only $n-1$ interactions. At the "high interaction" extreme, where every resource interacts with every other resource, there are $n(n-1)/2$ interactions. Typical examples at this extreme are medical surgery processes where intensive team collaboration is required. The IT incident management process may be positioned in the middle of this spectrum. Simple incidents may just need pairwise interaction during handoffs, but complex issues usually require intensive team interactions such as group chats.

This paper is organized as follows. In section 2 we discuss social network preliminaries. Section 3 gives applications where social networks play an important role. Our empirical study appears in Section 4. Section 5 describes how our approach can be used for resource allocation by solving optimization models. In Section 6 we generalize our methodology to other applications. Section 7 gives a literature review with a conclusion in Section 8.

II. SOCIAL NETWORK BACKGROUND AND NOTATIONS

Figure 1 shows an example of a social network. The nodes represent individual workers or resources, and the edges show the distances between connected pairs of nodes. An edge from user A to user B indicates that A and B know each other or have had interactions with each other. Note the edges are not directed. This means that each edge represents a symmetric relationship, i.e. if A knows B, then B knows A.

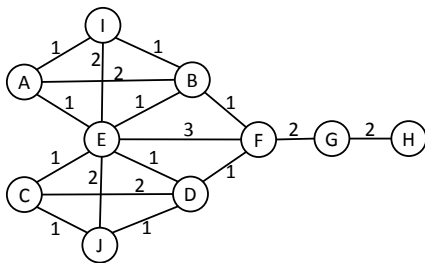


Figure 1. A conceptual graph representation of a social network

There are a number of metrics used to analyze a social network [22],[24]. We distinguish two types of metrics: network-based metrics that measure features of the network or its sub-networks; and local metrics which measure features of a particular node or edge in a network. Network-based metrics include size, density, cohesion etc. Typical local metrics include a number of node centrality measures, e.g. node degree, betweenness of a node etc. Some metrics, e.g. closeness, can be defined either at the network level or at the local level. In this paper, since we are interested in how resource team structure affects process performance, we will focus on network-based features. Next we define some of these metrics more formally and illustrate them using Figure 1. Given a network $G=(V, E, W)$ where V is the set of $|V|$ vertices (or nodes), E is the set of $|E|$ edges between a pair of vertices, and W is the weight of each edge.

- *Network Size* is the total number of vertices in the network. The size of the network shown in Figure 1 is 10.
- *Density* is defined as the ratio of the number of edges and the number of possible edges in the network, i.e. $\frac{2|E|}{|V|*(|V|-1)}$. The density of the network in Figure 1 is 0.38.
- *Closeness*: This metric indicates how near a node is to all the others. We first compute all shortest paths between all node pairs in the network and compute their average, i.e. $Avg_{i,j \in V}(d_{ij})$, where d_{ij} is the shortest path between vertices i and j . In Figure 1, the closeness of the network is 2.93.
- *Strength*: the strength of a vertex is defined as the sum of the edge weights of adjacent edges of this vertex. Also, network strength is defined as the total weights of edges in the network, i.e. $\sum_{e \in E} w_e$, where w_e is the weight of edge e . In our example the vertex strength for Node A is 4. The strength of the entire network is 25. However, in this paper, to make results more interpretable, we use weight to represent the distance between resources and *inverse of the weight* as the strength. In this way, the lesser the weight, the shorter is the distance between two resources, and the larger the strength between them.
- *Cohesion of network*: this metric is defined as the minimum number of nodes needed to be removed from the network such that the network gets disconnected. The cohesion in Figure 1 is 1, as removing node F or G will disconnect the network.
- *Clustering Coefficient (CC)* measures the degree to which nodes in a graph tend to cluster together. If node v has k_v neighbors, then at most $k_v(k_v - 1)/2$ edges can exist between them. If CC_v denotes the fraction of these edges that actually exist, then CC is the average of CC_v over all nodes. In Figure 1, $CC_A = 1, CC = 0.58$.
- *Hierarchy* measures the degree to which the network approaches a perfect hierarchy, i.e. degree to which all relations are unidirectional.

For local metrics, for example, *degree of a node* specifies with how many other nodes a certain node is connected; thus, node E would be the most privileged, with 5 connections. *Betweenness of a node* measures the number of shortest paths that pass through the vertex. In Figure 1, despite E having more direct connections, node F is a bridge, without which the other nodes would be disconnected. Hence, it is a single point of failure, and plays the role of information broker. It helps to keep the network unit intact. The normalized betweenness of node F is 0.42, and of node E is 0.32.

III. BUSINESS PROCESS SCENARIOS AND SOCIAL NETWORK FEATURES

In this section, we introduce examples from medical treatment and software development to illustrate the

importance of alignment between social networks and business processes and discuss a few significant social network features in each example that have been explored in literature. We also describe the incident management process on which our empirical study is based.

A. Medical Treatment Process

Health care is fundamentally about social interactions. Figure 2 shows a network of interactions among healthcare professionals in a hospital environment. In this network there are professionals like surgeons, anesthetists, nurses, operating department practitioners (ODPs), home care assistants (HCAs), and administrators. The directed edges show the flow of communications among the individuals involved. Thus, Nurse manager 1 receives a large number of communications as indicated by her large in-degree. Research has shown that patterns of communication among the operating team members can affect decision making patterns, lead to communication breakdowns and drastically affect patient outcomes [15],[23],[25],[30]. In particular the effect of density, clustering and hierarchy metrics on performance is discussed in [23],[25].

B. Software Development Process

Previous work [5],[6],[21],[28] has shown resource social networks have great impact on the quality of software. In most cases, resource social networks are an important factor to be considered during task assignment. Figure 3 shows development tasks in a software development project. These tasks are associated with each other due to shared artifacts (e.g. data tables, code, user interface etc.). To reduce dependencies and also improve efficiency, a common practice is to group tasks into clusters and try to assign a cluster of tasks to one resource or resources with strong ties. For the example shown in Figure 3 (a), these tasks may be grouped into three clusters, as highlighted in different colors. Moreover, some tasks, e.g. T6 and T7, may be identified as critical ones as they have high degree associations with others.

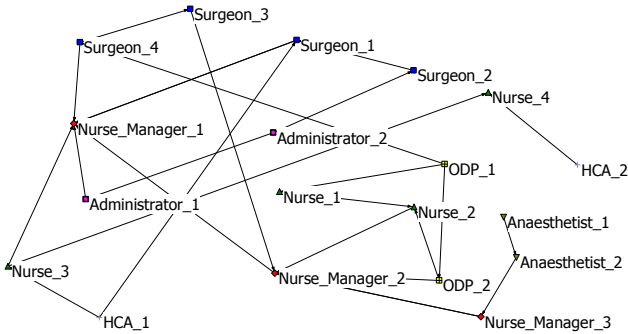


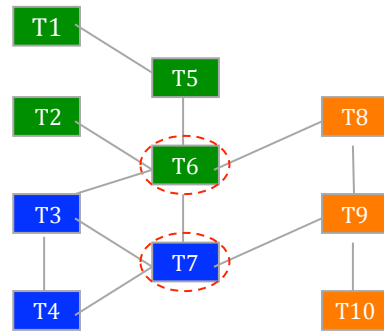
Figure 2. A healthcare professional social network

We can leverage social network features to optimize resource allocation. A community is a group of resources that is densely interconnected. Usually, resources within a community share common interests, e.g. database, BPM communities etc. Figure 3(b) shows the corresponding social network of resources in the software development project. These resources can be clustered into three communities as highlighted in different colors. For example, tasks T1, T2, T5, and T6, all related to data model development, can be assigned

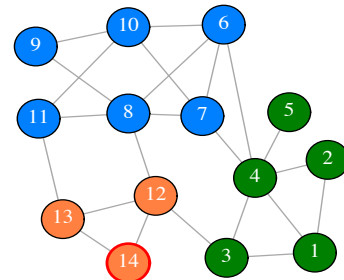
to resources in a database community consisting of resource nodes 1-5. In addition, previous studies [6],[21] showed that resource closeness, betweenness, and degree metrics are significantly correlated with the proneness to software failures. These metrics need to be considered during task assignment. Further, a critical task can be given to critical positions such as a key contributor, coordinator, or influencer [20] in the community, so they can use their social position effectively to improve process performance. Thus, task T6 may be assigned to node 4 as this resource is the center of the community and also bridges communication between communities. A formal method for resource allocation by utilizing social networks is discussed later.

C. IT Incident Management Process

Many enterprises outsource the management of their IT systems to professional IT service providers. When facing issues with the systems, business users of the IT systems submit incidents to a service provider. Appropriate resources from the service provider are then assigned to work on the incidents and resolve the issues within timelines as defined in a service level agreement. Typically, resources are organized into different support teams by their specialty, e.g., database, operation system, etc.



(a)



(b)

Figure 3. Software Development Tasks and Resource Community

Figure 4 illustrates an IT incident management process. A problem or issue faced by a business user is reported to a help desk. The help desk agent creates a new incident (task “Open Incident”) in a ticketing tool and records the description of the issue faced by the user. Then the incident is assigned to a specific support team based on the problem nature (task “Validate and Assign Incident”). The incident, once assigned to a support team, is picked up by an available resource within the team. The resource acknowledges the receipt of the incident by

claiming the ownership of the incident (task “Update Incident Ownership”) and then diagnoses it (task “Analyze Incident”). If needed, the resource may communicate with the customer for additional information.

However, the incident may be transferred to another resource in the same support team or to another support team (task “Transfer Incident”) for a number of reasons: (1) the incident may be misrouted due to insufficient information recorded; (2) the assigned resource may become unavailable; and/or (3) for the complexity nature of issues involved in the incident, multiple resources with different specialties may be required in the diagnosis and resolution. In such a case, the resource recommends an alternate resource to work on the incident or simply returns it to a dispatcher to determine the next appropriate resource. An incident is always assigned to one resource at a time. Once an incident is resolved, the resource restores the functionality of the system as required by the business user (task “Restore Service”). The business user validates and confirms the service provided by the resource (task “Confirm Service”). Once confirmed by the user, the incident is closed (task “Close Incident”).

In general, for the resolution of complex issues multiple resources with different specialties may be required in diagnosis and resolution. Incidents, particularly ones regarding application outages, are often complicated as their root causes may lie in multiple layers in the IT environments, including the application, middleware, hardware, network infrastructure, etc. Thus, it may be better if a system can identify in advance a series of resources that would work on different aspects of a ticket. This is more like assigning a team of resources to a ticket. Of course, some of them may not be available when they are needed. In such a case, the team is dynamically reassigned based on the state of the ticket.

It is clear from the above discussion that the various resources that resolve a severe incident work in a collaborative manner since a resource uses inputs provided by a previous resource in performing its task and may also consult with it to seek clarifications. Previous research has demonstrated the existence of such ties through social network analysis [20]. Furthermore, through empirical data, [20] shows that there are communities and workgroups in such a network and resources

serve social roles such as *contributor, influencer, coordinator*, etc. These roles are identified based on local metrics such as degree and betweenness. Roles such as influencer and coordinator are authorities and hubs that provide advice and connect other resources together. They serve as the glue in incident resolution. A follow-up question is what network-based metrics can be utilized during task assignment to improve performance. We continue with a formal analysis to answer this question.

IV. EXPERIMENT: DISCOVERING SIGNIFICANT SOCIAL NETWORK FEATURES

In the IT incident management process described above, each incident is handled by a group of resources including help desk agents and different support teams. We studied the process logs of 1562 incidents. The number of resources (excluding customers) involved in each incident ranges from 1 to 13, with an average of 3. IT incident management, as a knowledge intensive process, requires close collaboration between resources. The interactions between these resources, particularly, the interactions between support teams determines how efficiently a process instance can be executed. Next we use real case data to discover what specific features of resource social networks have significant impact on the process performance if they exist.

For IT incident management, the most critical key performance index (KPI) is resolution time (or Time to Repair, throughput time), i.e. the elapsed time between close time and open time of an incident. In our case, the elapse time also contains the waiting time for pending customer response (i.e. execution time of tasks “Provide More Information” and “Confirm Service” as shown in Figure 4). In practice, the waiting time is not counted in measuring the performance of incident processing since it is largely out of the control of support teams. Therefore, resolution time is calculated as the elapse time deducted by the waiting time. It is well known that many factors contribute to this KPI, including incident priority level, technology domain, incident complexity, performer skills etc. Typically, a service level agreement defines the timeline for resolving incidents at each priority level. Besides these well-known factors, here we conjecture that the resource social networks can be another contributing factor.

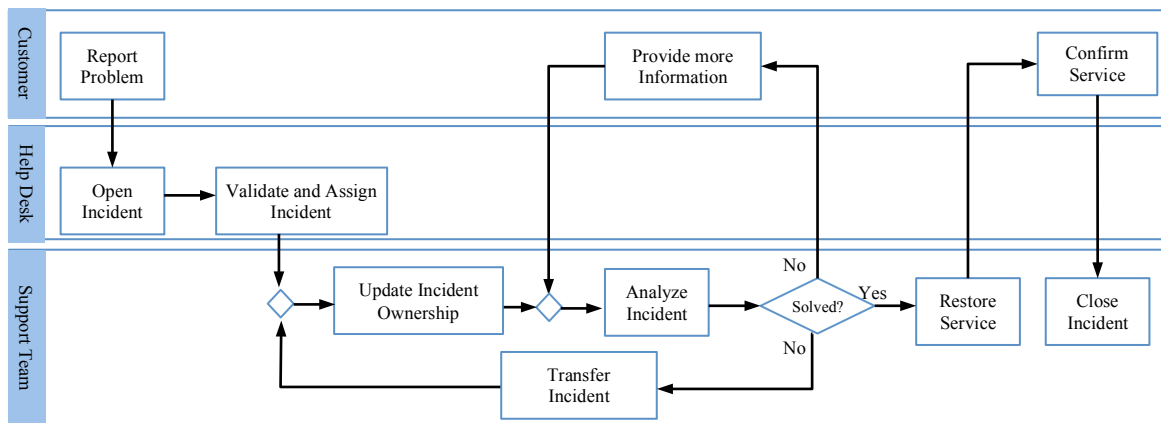


Figure 4. A process model for incident resolution

We collected 1562 closed incidents with detailed execution logs for each task and formulated a social network for resources (Help Desk agents and support teams) based on task execution logs. In this social network, each resource is represented as a node and a handoff between two resources as an edge. The *weight* associated with each edge is the *inverse* of the total number of handoffs between the two nodes. For example, if there are in total 5 incidents handed across two resources (in either direction) along the process flow, then the weight between them is 1/5. We use the weight to measure the distance between two nodes. When more handoffs occur between two nodes it implies they are closer. Other factors like execution time of consecutive tasks may also be considered in computing closeness, but we ignore them here. With this setup, we get a social network with 479 nodes and 1918 edges. The density of this social network is 0.017.

In this experiment, we try to determine features of the resource social network that significantly affect the performance of this incident management process, i.e. resolution time. For this experiment, ideally, we need to first estimate the effect of those well-known factors on this KPI, e.g. priority level, technology domain difference, and problem complexity. Unfortunately, information regarding these factors

is not disclosed in this data set. Thus, in order to isolate these factors as much as possible, we only consider a subset of incidents all processed by a key resource. For example, node 51 in Figure 5 worked on 158 incidents in total. We use the 158 incidents as our testing data set. By selecting all incidents that involved a single resource we can reasonably assume that these incidents were within the performer’s expert domain and they had similar technical issues. Also, resource 51 serves as a technical lead in the support team and it is also identified as an influencer and a coordinator within the resource social network. As a general practice in this process, a technical lead always gets involved in high priority or complex incidents. Thus, his participation may indicate these 158 incidents have relatively high priority and complexity. Thus, with this data selection process, we can assume that these incidents are similar in terms of complexity and priority, and it is reasonable to use them for our experiment.

The network constructed based on the 158 incidents is shown below in Figure 5. This network has 131 nodes and 602 edges with density 0.07. The degree of nodes ranges from 1 to 139 with an average of 9. Clearly, a few high degree nodes (i.e. with more edges) are placed in the center of the graph, surrounded by low degree nodes.

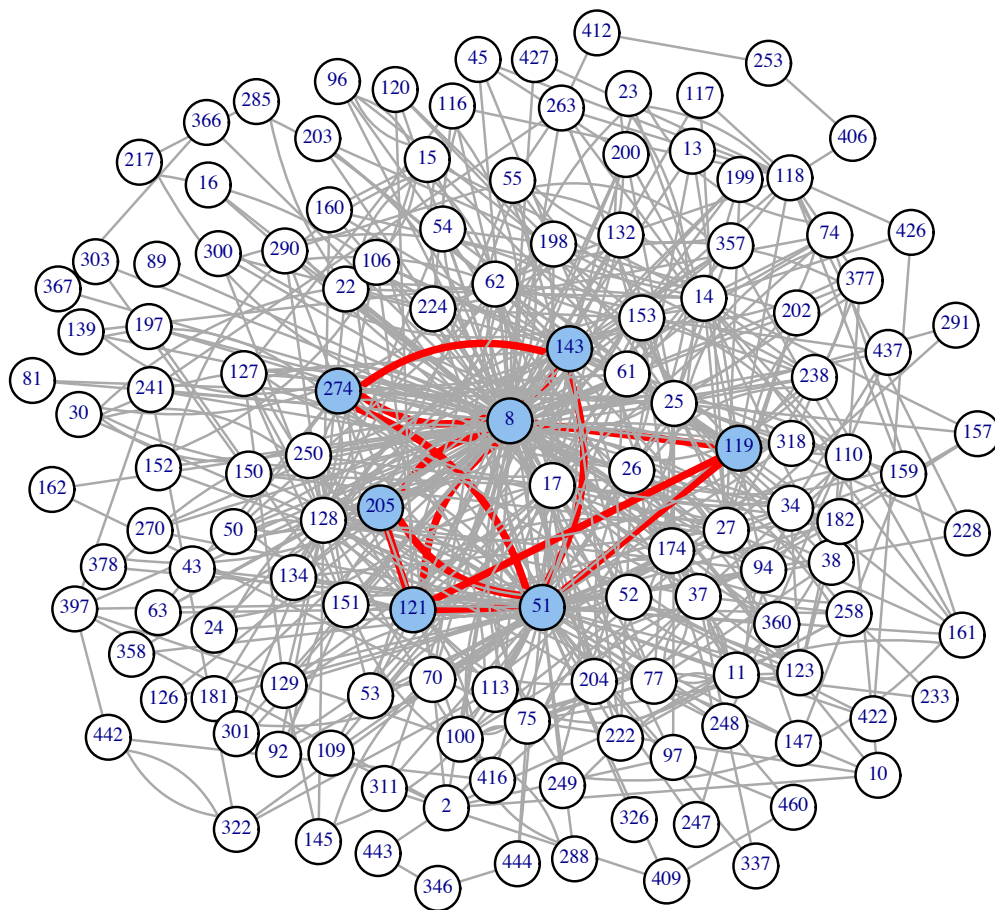


Figure 5. Resource Social network with 131 nodes and 602 edges of the IT Incident Management process showing all nodes that participate in process instances where resource 51 participates

In our case study, each incident is handled by a group of resources. For example, in Figure 5, we highlight 7 nodes (or resources) that collaborated on an incident. This incident was created by node 205, assigned by node 121 to a support team, picked by node 119 in the assigned support team, and transferred through a few resources from node 8 to 143 to 51, and then closed by node 274. This collaborative team forms a *sub network*. We are interested in exploring what features of the team, i.e. the sub network, significantly contribute to the performance of the incident resolution.

Hence, we calculate social network features for the sub network of resources that collaborate on each incident, not for the entire network (or a single node) of Figure 5. Previous work suggested that network size, density, centrality, cohesion and other metrics are correlated with resource performance [4]. Consequently, we choose *sub network size*, *density*, *strength*, *cohesion* and *closeness* as our candidate features and use linear regression to test if any of these features is significant. The definition of these features is given in Section 2. The summary statistics of these metrics for the sub networks corresponding to these 158 incidents are shown in Table I.

Table I shows that the mean resource count is 4.15, so four resources are involved in an incident on average. It takes averagely 103 hours or approximately 4 days to resolve each case. The mean density is 0.57 suggesting the typical sub network is moderately dense and 1.69 nodes must be removed to break it. The mean closeness indicates that the average shortest path length between any pair of nodes in a sub network is 0.41, no matter whether they are directly or only transitively connected. This short distance indicates resources in the sub network in general have a close relationship. The mean strength of 86.85 means that, on average there are about 87 handoffs among the resources in the sub network.

We first built a linear regression model with resolution time as the dependent variable and all candidate features as *features*, the number of resources (*res_count*) and closeness are

significant at the 1% and 5% confidence levels respectively, while the other features are not.

TABLE I. SUMMARY STATISTICS OF SOCIAL NETWORK METRICS

Metrics	Statistics					
	Min	1 st Qu.	Median	Mean	3 rd Qu.	Max
res_count	2.00	3.00	4.00	4.15	5.00	10.00
density	0.17	0.47	0.55	0.57	0.67	1.00
closeness	0.04	0.14	0.26	0.42	0.60	2.15
strength	1.00	22.00	68.00	86.85	145.00	334.00
cohesion	1.00	1.00	2.00	1.69	2.00	4.00
resolution time (hour)	0.20	4.77	27.41	103.40	96.90	2165.00

In the second round, we removed the insignificant variables and only kept the two significant ones to build the linear regression model again. The results shown in Figure 7 indicate that these two variables are still significant at confidence level 0.001. This experiment provides good evidence that the number of resources involved in a case and the closeness among them have strong impact on resolution time. It is intuitive that the number of resources matters because having more resources implies more handoffs and thus a longer throughput time. Also, due to the high complexity of the incidents in this data set, an incident owner frequently uses group chats with all the people that have been involved in the case to resolve the incident.

The effectiveness of such team interactions largely depends on the social distance among the resources, which can be measured by closeness of the sub network. This finding can also be demonstrated through a scatter plot as shown in Figure 8. Clearly, a large portion of points follow the plane defined by the regression model, but some points are far from it. This can be explained by the fact that resolution time is determined by many factors, not just social network features. Certainly, we cannot predict the resolution time of each case solely by social network features. In this experiment, our intention is to find social features that significantly affect process performance.

```
lm(formula = resolution_time ~ res_count + density + closeness +
  strength + cohesion, data = tkt)

Residuals:
  Min      1Q  Median      3Q      Max
-424.13 -84.66 -46.58  15.90 1494.58

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -395.7483   149.0344  -2.655  0.00877 **
res_count    64.5227    23.9372   2.695  0.00782 **
density     340.9902   192.0203   1.776  0.07777 .
closeness   190.2589    75.4942   2.520  0.01276 *
strength     -0.3762     0.5798  -0.649  0.51737
cohesion     -5.1628    39.2311  -0.132  0.89547
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 234.8 on 152 degrees of freedom
Multiple R-squared:  0.1446,    Adjusted R-squared:  0.1165
F-statistic:  5.14 on 5 and 152 DF,  p-value: 0.0002196
```

Figure 6. Results of linear regression with all variables

```

lm(formula = resolution_time ~ res_count + closeness, data = tkt)

Residuals:
    Min       1Q   Median       3Q      Max
-362.50  -96.80  -32.54   15.54 1592.11

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -114.78     53.41  -2.149  0.033183 *
res_count     38.21     11.27   3.390  0.000885 ***
closeness    142.13     47.76   2.976  0.003386 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 235.7 on 155 degrees of freedom
Multiple R-squared:  0.1211,    Adjusted R-squared:  0.1097
F-statistic: 10.68 on 2 and 155 DF,  p-value: 4.532e-05

```

Figure 7. Results of linear regression after removing some variables

Note that although we have discovered two significant social network features through the experiment, this finding only applies to the IT incident management process described in this paper. It may not always be the case for other business processes, but similar experiments can be performed to discover such features. Next, we discuss how to leverage these significant features in task assignment.

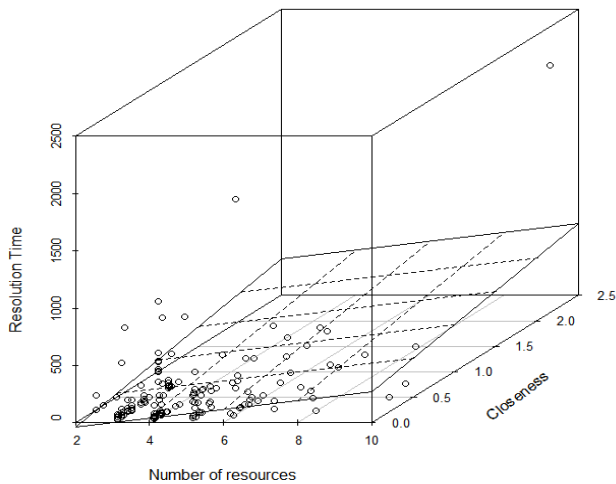


Figure 8. A 3-d plot of resolution time as function of two variables

V. MODELING OPTIMAL RESOURCE ASSIGNMENT

A general approach to modeling resource assignment is to optimize with respect to the key social network features that are found by analyzing actual data from the network. If there are several metrics, the objective function is a weighted sum of each metric. We illustrate this approach in the context of incident management and software development.

A social network is a graph (see Figures 1 and 2) where each node represents a resource. Previous research [20] has shown that this graph can be organized into communities, and communities in turn have work groups and roles. Thus, we can label a resource node r_i as:

$$(Tech_Role_i, Com_i, Grp_i, Soc_Role_i)$$

where

i is the node id

$Tech_Role$ is the technical role the resource plays

Com_i is the community node i belongs to

Grp_i is the workgroup within the community i belongs to

Soc_Role_i is the social role of i within Com_i or Grp_i .

Table II shows a few resources along with communities, workgroups and social roles identified in the previous work from real incident management data [20]. For example, resource 51 has “Support Team” as her technical role, she belongs to “Procure to Pay” workgroup in the community of “Purchasing & Payment”, and plays a number of social roles as an influencer and a coordinator of this community and a key contributor of “Procure to Pay” workgroup.

For the IT incident management process, based on our experiment, in assignment of a resource team, the goal is to find a subset of resources R_{sub} as a team from the available resource pool R that has the smallest closeness (denoted as $C(R_{sub})$) among all feasible teams. This is expressed as:

$$\text{Min}_{R_{sub} \subseteq R} C(R_{sub}),$$

such that

for each task t , there is an assigned resource $r_t = (Tech_Role_t, Com_t, Grp_t, Soc_Role_t)$ that matches the task technical role, community, workgroup and social role requirements.

We did not include resource count in the objective function because it is implied by closeness. Moreover, the constraints will specify the minimum number of resources required. Typically, technical role constraints are hard, while the social role constraints are soft and can be relaxed. Thus, we may relax the social constraints if the only suitable technical role that is available does satisfy them, or is from a different community or group than the one desired. Next we use a hypothetical example to illustrate this model. Assume that when a customer reports an issue, only the resources listed in Table II are available. This process shown in Figure 4 requires two technical roles, Help Desk and Support Team. Hence resource 205 is assigned to execute “Open Incident” and “Validate and Assign” tasks. Also, based on the social network shown in Figure 5, among these four support team resources, resource 121 is the nearest one to resource 205. So resource 121 is assigned to execute all the support team tasks in the process.

When resource 121 has to transfer the incident, a new support team resource needs to be added. By the optimization model, this new node should be closer to *both* nodes 121 and 205 than any other node. In other words, among all triad sub networks consisting of nodes 121 and 205, the one with smallest closeness is optimal. In this case, resource 51 is recommended. Note this illustrative example is different than the actual case highlighted in Figure 5.

This is a NP-complete assignment problem. A heuristic would find an initial pair of nodes with the minimum shortest distance between them. In subsequent steps it would enlarge this subset by adding nodes to it such that additional nodes have the shortest distance to the current subset of nodes until all the technical resource requirements are satisfied.

Similar formal optimization models can also be used to assign resources in software development process. For example, based on previous work [6],[21], closeness, betweenness and degree metrics are significant social network features that can be used to predict software failures. Thus, we can formulate a task assignment model with the goal as finding a sub network (i.e. a group of resources) that has minimal weighted sum of these metrics, subject to the constraint that it matches the role requirement of each task in a process instance. Thus:

$$\text{Min}_{R_{sub} \subseteq R} \lambda_1 C(R_{sub}) - \lambda_2 \sum_{r \in R_{sub}} \text{Bet}(r) - \lambda_3 \sum_{r \in R_{sub}} \text{Deg}(r)$$

where

$C(R_{sub})$ is the closeness of the sub network R_{sub} of resources assigned to tasks,

$\text{Bet}(r)$ is the betweenness of resource r in the sub network

$\text{Deg}(r)$ is the degree of resource r , and

$\lambda_1, \lambda_2, \lambda_3$ are weights of each metric

These models can be solved using optimization software like CPLEX [16]. Initially, we can analyze historical process execution logs of the resource social network. If such data does not exist, we start with default values which may be revised periodically until the system reaches a steady state. When allocating resources for a particular process instance, the resources are first filtered based on skill, permission, availability and other hard constraints. Then the proposed models are solved to get a subset of resources that form a team for this instance. Often, the team may change dynamically based on resource availability, and the model can be solved again at an intermediate stage as needed to find new solutions.

VI. DISCUSSION

We presented an approach for identifying social network metrics that affect throughput time of a process instance and for modeling an optimization problem that can be solved to

determine an appropriate resource assignment in a given environment. While there is agreement that process performance improves when the resources that participate in a collaborative process or modules of a software package are close to each other in a social network it is not clear what specific metrics of the social network play a role in delivering the improved performance.

In the incident resolution process we found from our empirical study that resource count and closeness were two significant factors that had an impact on throughput time. On comparing results of social network analyses in other domains such as medical treatment and software development, it appears that these factors are not the same across all applications. Each application has a different degree and pattern of team interactivity, and a different mix of social network metrics is needed to capture it. The incident resolution process has some unique aspects that are different from those of other team processes. Here the interactions tend to be predominantly between successive resources that work on a case. Hence, the closeness among successive resources or a subgroup of resources is more important than the interactions among all resources on a team that works on an incident. This is reflected in the closeness metric.

Further, the notion of a team is different from that in, say, a medical surgery process where interactions among all the team members are denser. Another difference between these two applications is that in incident resolution although an initial assignment of resources for an incident is made upon incident arrival, *these assignments can change dynamically as the case progresses based on the availability of a certain resource* at the time it is required and also the assessment of a resource working on the case as to who the next best resource would be. In contrast, in a surgery all or most resources have to be available for the entire duration of the procedure from start to finish.

Clearly, it is hard to draw generalizations from a single empirical study and that is not our intention here. Rather it is to recognize and emphasize that different social network metrics influence the performance of cases in different application settings. By recognizing this observation, we have tried to associate different metrics with various applications where these metrics would be appropriate based on our understanding of the nature of interactions required in different applications and the results reported in literature. In the incident resolution case, group interactions between a pair or among a subset of the resources involved in a case are often an effective way to resolve complex incidents; hence, it is helpful to minimize the closeness of the sub-network. On the other hand, in surgery all surgical team members interact with one another.

TABLE II. RESOURCE EXAMPLES IN IT INCIDENT MANAGEMENT PROCESS

Resource	Technical Role	Community	Workgroup	Social Role
51	Support Team	Purchasing & Payment	Procure to Pay	Coordinator, Influencer, Key Contributor
55	Support Team	Travel & Expense	Security	Key Contributor
119	Support Team	Purchasing & Payment	Customer Application	Influencer, Key Contributor
121	Support Team	Purchasing & Payment	Procure to Pay	Influencer
205	Help Desk			

Hence, the density becomes an important metric [23], [25]. In software development resources who work on related modules must interact very closely. Their collaboration is tighter than that among resources in incident resolution. Hence, the closeness, betweenness, degree properties are critical [6],[21]. We also conjecture that centralization and hierarchy properties are more important where a single or a small group of resources are involved repeatedly in completion of a process, as in a strategic management process where, say, a CEO or some key people are intimately involved throughout various stages. This conjecture is supported in part by [3].

Finally, some processes are modular and consist of sub-processes. Within each sub-process the interaction is strong, but across sub-processes it is weaker. An example would be the order-to-cash process where different departments like sales, production, logistics and finance are involved in that order, and each performs its own subprocess. In this case, our hypothesis is that clustering coefficient may have significant impact on the process performance.

Some limitations of our work are as follows. First, our data for the regression experiment assumed that all incidents in which a key influential resource was involved would be of similar complexity. Incident complexity varies widely and it is very hard to predict in advance how complex an incident may turn out to be. However, the fact that it passed through a certain important individual suggests that it exceeded a threshold of complexity. Second, there is a large number of social network metrics. We considered several social network metrics that we perceived to be important in the context of incident resolution in our experiment but we could not consider all of them. Thirdly, our results about significant social network metrics apply to only one domain and the extensions to other domains are conjectures that need to be verified further.

VII. RELATED WORK

The evolution of socio-technical systems dates back to the work of Trist [29] and refers to the interrelatedness of social and technical aspects of an organization. Sociotechnical theory focuses on joint optimization [8] that is, designing the social system and technical system in tandem so that they work smoothly together.

Inspired by this early work, there is a new wave of interest in leveraging social computing in business process management, and particularly for allocating tasks of a process instance to resources that are mutually compatible [19]. There have been some initiatives in this line of research. First, researchers have analyzed process execution logs to understand resource social behaviors [1],[12],[20],[27]. [1] introduces a systematic approach and also a tool called MiSoN that creates social networks based on task execution logs. [12] gives rules and algorithms for finding resource interaction patterns in ad-hoc processes. A social network analysis for IT Incident management process was conducted in [20]. The idea of extracting social networks through organizational mining of process logs is introduced in [27] and algorithms for doing so are discussed.

Second, social BPM studies how to fuse BPM practices with social networking applications to improve process

performance. Studies in this area [7],[10],[13],[17] propose the integration of different social tools, e.g., instant messaging, Wikis, discussion forums, to engage stakeholders either inside or outside an organization in designing process models or collaborating on task execution. [18] proposed a system architecture and modeling notations in support of social BPM. Another concept in Social BPM is the Social Compute Unit (SCU) [11],[26], a collaborative resource unit formed dynamically during task execution. Our work aligns well with this concept. However, in contrast to the previous approach [[26]] that identifies SCU based on task technical dependencies alone, here we proposed to leverage significant social features also to identify such optimal teams with regard to process performance.

In addition, in the literature, there is a large body of work on the role of socio-technical networks in software development. Inspired by the original Conway's law [9], many studies continued to understand social network effects on resource performance [5],[6],[28]. By first building a dependency network among software modules and a contribution network among resources that contribute to modules, [6] showed by means of a predictive model that socio-technical features influence the failure-proneness, an important KPI, of the software. In [5], the authors give evidence to suggest that better communication among developers led to faster bug resolution. The collaboration patterns among developers are also studied elsewhere [28] suggesting they are important, although performance implications have not been evaluated.

Socio-technical networks play a major role in healthcare. In [23], it is shown that distinct patterns and structures of communications emerge among the various medical teams' leadership based on their hierarchy and density metrics, and they affect the efficiency and effectiveness of team function. In [15] it is shown that, "serious communication breakdowns occur across the continuum of care, typically resulting from a failure in verbal communication between a surgical attending and another caregiver..." A social network analysis of two primary care practices [25] found widely differing decision making patterns as reflected in their respective network density, clustering coefficient, and centralization metrics. They hypothesize that network density would correlate positively with practice adherence to treatment guidelines, and network hierarchy would correlate negatively. The work in [30] showed that simply use of a checklist (which is similar to a process) by the senior surgeon during the conduct of a surgical operation resulted in significantly higher team communication scores in the surgical group.

VIII. CONCLUSION

While there is evidence to show that patterns of communication in a social network affect decision-making behavior and performance of a team process, it is not clear what specific metrics of the social network are important in different domains. Our empirical study using data from an incident resolution process confirmed that social network features do affect the performance in terms of resolution time, and identified the number of resources and closeness as the two metrics that affect performance in a statistically significant

way. We also found in our study that other factors such as strength, density and cohesion were not significantly related to throughput. It should be noted that the influence of these factors on performance is not the same across all applications and depends upon the specific nature of the processes involved and the patterns of interactions among the collaborating resources required to complete them.

Thus, we have developed models for optimizing resource assignment to tasks in a team process taking into account the nature of their interactions as reflected in the social network. Our methodology consists of two parts, where in the first part empirical data is used to determine important social network metrics and in the second part a model is applied to enable optimal resource allocation. In this paper, we tested the first part of the methodology through an empirical study to show that some social network metrics are more important than others. However, we leave the testing of the models described in the second part of the methodology for future work. Suffice it to say that the models need to be application domain and process specific, and the same models cannot be used in all situations.

In future work it would be helpful to implement these models and evaluate their performance. We would also like to collect more data from processes in the incident resolution and other domains to validate our view that the relative importance of different metrics varies in different domains.

Reference

- [1] Aalst, W. M. P., Song, M.: Mining Social Networks: Uncovering Interaction Patterns in Business Processes, *Business Process Management*, 244-260 (2004)
- [2] Amrit, C., Hillegersberg, J., and Kumar, K.. "A social network perspective of Conway's law." *Proceedings of the CSCW Workshop on Social Networks*, Chicago, IL, USA. 2004
- [3] Balenzano, C. , Scardigno, A. F. , Giancaspro, M. L. , Manuti, A. , & Jacobone, V. (2013). Social Network Analysis and Strategic Planning: A Case Study from the Italian Local Context. *American Journal of Applied Psychology*, 1(2), 26-32.
- [4] Balkundi, P. and Harrison, D. Ties, Leaders, and Time in Teams: Strong inference about network structure's effects on team viability and performance, *Academy of Management Journal*, Vol 49(4), 2006
- [5] Bernardi, Mario Luca, et al. "Do developers introduce bugs when they do not communicate? the case of eclipse and mozilla." *Software Maintenance and Reengineering (CSMR)*, 2012 16th European Conference on. IEEE, 2012
- [6] Bird, C., Nagappan, N., Gall, H., Murphy, B., and Devanbu, P. 2009. Putting It All Together: Using Socio-technical Networks to Predict Failures. In *Proceedings of the 2009 20th International Symposium on Software Reliability Engineering (ISSRE '09)*
- [7] Brambilla, M., Fraternali, P., and Ruiz, C. 2012. Combining social web and BPM for improving enterprise performances: the BPM4People approach to social BPM. In *Proceedings of the 21st international conference companion on World Wide Web*, 223-226
- [8] Cooper, R., Foster, M. (1971). Sociotechnical systems. *American Psychologist*, 26, 467-474.
- [9] Conway, M. "How do Committees Invent?," *Datamation*, vol. 14, no. 4, pp. 28-31, 1968
- [10] Dengler, F., Koschmider A., A. Oberweis, and Zhang H. Social software for coordination of collaborative process activities. In *Third Workshop on Business Process Management and Social Software*, LNBIP, pages 396-407, 2010
- [11] Dustdar, S. and Bhattacharya, K. (2011). The Social Compute Unit. *IEEE Internet Computing* 15(3): 64-69
- [12] Dustdar S. and Hoffmann T.. 2007. Interaction pattern detection in process oriented information systems. *Data Knowl. Eng.* 62 (1): 138-155
- [13] Erol, S., M. Granitzer, S. Happ, S. Jantunen, B. Jennings, P. Johannesson, A. Koschmider, S. Nurcan, D. Rossi, and R. Schmidt. Combining BPM and social software: contradiction or chance? *Journal of Software Maintenance and Evolution*, 22:449-476, 2010
- [14] Gawande, Atul. *The checklist manifesto: how to get things right*. Vol. 200. New York: Metropolitan Books, 2010
- [15] Greenberg, Caprice C., et al. "Patterns of communication breakdowns resulting in injury to surgical patients." *Journal of the American College of Surgeons* 204.4 (2007): 533-540
- [16] ILOG: Ilog CPLEX software, Version 11.010, 2008
- [17] Johannesson, P., B. Andersson, and P. Wohed. Business process management with social software systems-a new paradigm for work organisation. In *Business Process Management Workshops*, LNBIP, pages 659-665. Springer Berlin Heidelberg, 2009
- [18] Koschmider, A., M. Song, and H. A. Reijers. Social software for modeling business processes. In *First Workshop on BPM and Social Software*, LNBIP, 2009
- [19] Kumar, A., Dijkman, R. M., Song, M. Optimal Resource Assignment in Workflows for Maximizing Cooperation. *BPM 2013*: 235-250
- [20] Liu, R., Agarwal, S., Sindhgatta, R. R., Lee, J. Accelerating collaboration in task assignment using a socially enhanced resource model, *BPM 2013*: 251--258, Springer, 2013
- [21] Meneely, A., Williams, L., Snipes, W., and Osborne, J. 2008. Predicting failures with developer networks and social network analysis. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering (SIGSOFT '08/FSE-16)*.
- [22] Newman, M.E.J. 2010. *Networks: An Introduction*. Oxford, UK: Oxford University Press
- [23] Palazzolo, M., et al. "Measuring social network structure of clinical teams caring for patients with complex conditions." *Procedia-Social and Behavioral Sciences* 26 (2011): 17-29.
- [24] Package igraph, <http://igraph.sourceforge.net>
- [25] Scott, John, et al. "Social network analysis as an analytic tool for interaction patterns in primary care practices." *The Annals of Family Medicine* 3.5 (2005): 443-448
- [26] Sengupta, B., Jain, A., Bhattacharya, K., Hong-Linh Truong, H. L., Schahram Dustdar, S.: Who do you call? problem resolution through social compute units, *Proceedings of the 10th international conference on Service-Oriented Computing (ICSOC'12)*, page 48-62, 2012
- [27] Song, M., and Wil MP van der Aalst. "Towards comprehensive support for organizational mining." *Decision Support Systems* 46.1 (2008): 300-317
- [28] Surian, D., Lo, D. & Lim, E.-P. (2010). Mining Collaboration Patterns from a Large Developer Network.. In G. Antoniol, M. Pinzger & E. J. Chikofsky (eds.), *WCRE* (p./pp. 269-273), : IEEE Computer Society. ISBN: 978-0-7695-4123-5
- [29] Trist, E., & Labour, O. M. o. (1981). The evolution of socio-technical systems: A conceptual framework and an action research program: Ontario Ministry of Labour, Ontario Quality of Working Life Centre
- [30] Turrentine, B., et al. Studying Communication Patterns during Surgery. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 47. No. 12. SAGE Publications, 2003