Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2012 Proceedings | Proceedings

Modeling Service Interaction Networks

Zamani Leila
School of IT, The University of Sydney, Sydney, NSW, Australia., leilaz@it.usyd.edu.au

Joseph Davis
School of IT, University of Sydney, SYDNEY, NSW, Australia., joseph.davis@sydney.edu.au

Modeling Service Interaction Networks

Leila Zamani
The University of Sydney
Leilaz@it.usyd.edu.au

Joseph G. Davis
The University of Sydney
Joseph.Davis@sydney.edu.au

ABSTRACT

Service systems rely on internal interactions of service provider agents and the external interactions with customers in the design and delivery of services. Careful analysis and modeling of such interactions are essential to the design of effective service systems. This research focuses on service interaction networks in the context of the design and delivery of information technology (IT)-centric services. We develop and test a model of service interaction network effectiveness and investigate the effects of some of its structural properties on the effectiveness of service systems. We empirically investigate the validity of the model by using data from SourceForge.net and develop and test a set of specific hypotheses. The results indicate that network centrality and the network density have negative impacts whereas network size has positive influence on service systems effectiveness.

Keywords

Service interaction networks, service systems, social networks.

INTRODUCTION

The need to integrate organizational and human understanding with technological support to address some of the fundamental problems in service systems such as improving service productivity and quality has been widely recognized (Maglio, Srinivasan et al., 2006; Maglio and Spohrer, 2008). Increasing the competence levels of both the provider and customer side of service systems are critical to enhancing the performance of these systems. As well, service systems rely on internal interactions of service provider agents and the external interactions with customers in the design and delivery of services. Careful analysis and modeling of such interactions can be helpful for the design of effective and efficient service systems.

In a typical service delivery system, each service request is performed by a group of actors based on a set of negotiated requirements. This setting of service delivery is viewed as an abstract network and the collection of agents representing various tasks and their interactions and the outcomes is referred to as a *service interaction network* (Kameshwaran et al., 2009). Analyses of service interaction networks (SIN) help to model a wide range of issues that arise in such a setting. In studying this type of service systems, it is necessary to consider the following aspects: 1) how do the network's agents (providers and customers) interact to establish an efficient and effective service delivery system, and 2) how do individual competence and structural patterns come together to create value across the system.

This paper takes into account these two aspects in studying SINs in the context of the design and delivery of IT-centric services. More specifically, it proposes a model of SINs in service delivery and investigates the critical factors that characterize the interactions among the network agents as they impact on the performance of the system. This model is based on the idea that certain structural properties of SINs can have significant effect on the effectiveness of the service system. This paper aims to a) provide a better understanding of the role of service provider and service customer in SINs, and b) examine the structural properties of SINs that can impact service delivery effectiveness.

To the best of our knowledge, this research is the first to study the effectiveness of service systems by modeling the structural aspects of SINs to explain such service system effectiveness. In order to examine the validity of the proposed model, we test it with real world data. One of the key research challenges in this domain of service systems is the sensitivity and proprietary nature of the real world data on service systems (Kameshwaran et al., 2009). We use data from Open source software (OSS) projects which provide reasonable proxies for the interactions between service provider and customer agents, which roughly correspond to developers and users, to fix bugs or develop new features. We analyze this data using a sample of OSS projects to test a number of specific hypotheses suggested by the proposed model.

This paper is organized as follows. The next section reviews the literature of SINs and provides the theoretical background. In section 3, the effectiveness model for SINs is proposed. In the following section, several hypotheses based on the proposed model are developed and discussed. Data and methodology to test the hypotheses are elaborated in the fifth section. Following this, the analysis and results are reported and discussed. Our conclusions and directions for future research are included in the last section.

REVIEW OF THE LITERATURE

Social structures of SINs play a critical role in the performance of service delivery systems. Based on social network theory, we study the interaction patterns among service providers and service customers in order to investigate their impact on the effectiveness of the system. In this section, we review the concept of SINs and related theoretical background which lead us to clarify the research gap considered in this paper.

Service Interaction Networks (SIN)

In order to achieve the overall goal of any service system, customer's needs must be met. A useful approach to meeting customers' changing needs is for providers to interact closely with them to update and adjust the service delivery goals, processes, and outputs accordingly. Furthermore, different aspects of these interactions such as their effects on the whole system, and the effectiveness of their inter-relations must be taken into account.

Kameshwaran et al. (2009) analyzed interaction patterns in service delivery systems to understand the impacts of individuals and their interactions on the effectiveness of the whole system. The presence of some issues like confidentiality, lack of visibility or lack of control (in some cases) in service organizations make it unlikely to capture all micro level details of implementation including individual agent performance, all interaction patterns, etc. The notion of service interaction networks (SIN) was introduced by Kameshwaran et al. (2009). It helps to investigate and analyze the wide range of practical problems that arise in service delivery through a formal setting. In SIN there is a set of agents A= {1, 2, ..., N} who are involved in accomplishing assigned tasks (services) T= {1, 2, ..., T}. The agents interact through the network to perform the tasks and achieve the outcomes. However, specific tasks may require repeated interactions which are represented as direct edges in the network.

Prior studies on SINs have focused on developing new algorithms to examine agents' attributes individually and in the team, and mostly consider the issues of ranking agents in a service organization (Kameshwaran et al., 2009; Dixit, Kameshwaran et al., 2009b; Kameshwaran, Pandit et al., 2010). They used features like success, failure or cost of a task workflow to develop their iterative algorithm for ranking agents in service delivery systems. Dixit et al. (2009b) considered three factors to develop their agent ranking algorithem in inetraction networks: *1)* number of past interactions, *2)* rank of other agents with whom he interacted, and *3)* outcomes of the interactions. Dixit, Kameshwaran et al. (2009b, 2010) argued that service organisations can use such agent ranking and analysis to form more effective and efficient teams.

In a related study (Kameshwaran et al., 2009; Dixit, Kameshwaran et al., 2009a), a simulation platform was developed to generate data for SINs that are likely to reflect real life observations. The main motivation for simulating SINs was the sensitive nature of the required data. Simulation platform helped to analyze the degree of human interactions in service delivery and evaluate the impacts of those interactions on total service delivery time and the overall quality (Dixit et al., 2009a).

Prior studies have mostly focused on studying agents' interactions and their impacts on the efficiency of whole system. They have typically investigated how the agents can influence success or failure of the system. However, understanding other behavioral and structural factors which can impact on the system performance is critical to the design of effective service systems. SINs literature has not focused on how networks attributes (e.g. structural patterns of the networks) can cause success or failure of the service delivery system, and how structural patterns of the networks can be applied to form more effective team through the SIN. In particular, our research addresses this gap in the literature and proposes and tests a model of SIN.

SERVICE INTERACTION NETWORK EFFECTIVENESS MODEL

Although the previous studies considered the outcomes of SINs, they rarely address the determining influences on the performance of the systems. Our goal is to develop a model of SINs in service delivery and to investigate how structural patterns of networks can affect the effectiveness of service systems.

In following sub-sections, we define the indicators that capture the structural patterns of SIN in our model, as well as the measures to represent SIN effectiveness.

Measures of SIN Structural Patterns (Independent Variables)

Based on the similarities between SINs and social networks, we employ social network analysis (SNA) approach to identify the distinct patterns in such networks and explore the relationships between the structural variables and the outcomes of SINs. Prior research suggests that social network measures are useful indicators of various group outcomes (Coleman, 1988; Wasserman and Faust, 1994; Carrington, Scott et al., 2005; Hinds and Lee, 2008). Kidane & Gloor used SNA to discover the correlation between social structure of the team and the final products in software systems (Kidane and Gloor, 2007). In other work, Wu et al. (Wu, Goh et al., 2007) employed SNA to examine the relations between the success and communication patterns of the development team in open source software (OSS) projects. Studying social structure of the team helps better understanding of team practices like coordination, control, socialization and learning (Freeman, 1978; Scacchi, 2002). The social structure of SINs describes how providers and customers interact, communicate, and organize in the system.

The interaction patterns of SIN can be characterized by several attributes. They are used as independent variables in our proposed model. In social network analysis, the group density and centrality are related to the structure of the group, leadership, and the levels individual satisfaction. These two measures describe the level of criticality around particular nodes in the network and they are considered as two practical measures to study the communication patterns of a network (Scott 2002).

a) *Network Density:* Network density is defined as the quantity of lines in a network graph and it is measured as a proportion of the maximum number of lines (Scott, 2000). Network density score summarizes the overall number of interactions in the network. It is common to use network density in studies which consider the communication patterns of the networks (Crowston, Annabi et al., 2003; Crowston and Howison, 2005; Wu et al., 2007; Hinds and Lee, 2008).

b) *Network Centrality:* There are various standard measures of centrality in the literature (Wasserman and Faust, 1994) and choosing appropriate measure for centrality depends on the nature of networks interactions. In this research, two types of centrality are examined which are *In-degree* centrality and *Out-degree* centrality. The *in-degree* centrality for an individual node is the number of direct links received from other nodes, and the *out-degree* centrality is the number of direct links sent to other nodes. In SINs, an agent with high in-degree is distinguished as relations to it are important for a greater number of agents. An agent with high out-degree is significant as it can make its opinions known to a greater number of agents.

   In our study, we do not focus on the individual agent's centrality, but on the centrality of the whole network. The whole network centrality captures the inequality of the contributions between the agents: high score of in-degree centrality implies that the power of agents to be contributed by other agents varies rather substantially; the same approach is applied for out-degree centrality. Overall, high degree centrality for whole network describes that positional advantages are unequally distributed in the network (Wu et al., 2007). We examine if there is a relationship between this unequally distribution and the outcomes of the system.

c) *Network Size*: Network size is another indicator that can explain the structure of the networks. There may be more interactions between agents in larger networks. Previous studies in software systems show that bug resolution time is different among various team size (Au, Carpenter et al., 2009). In our model, network size is measured by the number of core agents (providers and/or customers) in whole network which is called *team size* in this model. We investigate the potential impacts of this measure on the effectiveness of SINs.

Measures of SIN Effectiveness (Dependent Variables)

There is general consensus that integrating services through a network of providing agencies (instead of separated organizations) will lead to a more effective system and more positive outcomes (Turrini, Cristofoli et al., 2010). Although the network effectiveness theories tend to operate at a more abstract level, we identify the key proxies of network effectiveness and identify the measures of effectiveness as dependent variables in our proposed model. In this study, we consider network effectiveness from the service provider side.

a) *Service Delivery Time*: Provan and Milward (1995) defines network effectiveness mainly at the client level. They argue that the overall quality of service delivery may improve the network effectiveness. This definition has been considered by many researchers who have empirically tested network success in various fields (e.g. Family and children services (Page, 2003); Job and training networks (Jennings and Ewalt, 1998) and community care network (Milward and Provan, 2003)). In our model, 'average service delivery time' is studied as a related measure to the quality of service delivery from provider side. The service delivery time for each requested service refers to the time which elapses between receipt of user orders and the delivery of the service. Some literature on OSS success measurement suggests that time to fix bugs (or implement new features) may be a useful measure of project success (e.g. (Crowston et al., 2003)). In addition, the total order cycle time is described as a performance measure in some delivery systems like supply chain systems (e.g. (Gunasekaran, Patel et al., 2001)). Thus, in our proposed model the average service delivery time is the first indicator to measure the effectiveness of SINs.

b) *Service Delivery Activity:* Provan and Milward (2001) expanded their network effectiveness model and add two more levels (community level and network level). Turrini et al. (2010) proposed an integrated framework of network effectiveness based on previous models (Provan and Milward, 1995; Provan and Milward, 2001) and classified network level effectiveness into 'network capacity of achieving stated goals' , ' network sustainability and viability' and ' network innovation and change'. Achieving stated goals can be measured by the rate of service delivery activity in a process. Based on the above, service delivery activity rate can be considered as an indicator to measure the network effectiveness.

Wu et al. (2007) examined development activity as a key indicator of OSS project success. They calculated this measure as the number of closed bugs in a fixed period. Ewusi-Mensah (1997) argued that completing project may be a sign of success and effectiveness in information systems. Crowston and Scozzi (2002), instead, measured OSS project success using a measure of project progress. In our study, service delivery activity is represented by *task completion rate* which measures by the number of completed jobs or closed requests over a fixed period.

HYPOTHESES

Given the objective of this research, particularly to investigate the impacts of structural patterns of the networks on the SINs effectiveness, we develop our model (and related hypotheses) of some of the key determinants of SIN effectiveness. As discussed in section 3, network density, network centrality and network size are defined as indicators to capture the structural patterns of SIN (independent variables). In addition, we characterized service delivery time and service delivery activity to measure the effectiveness of such networks (dependent variables). The research model and the respective hypotheses (H1 to H6) are presented in Figure 1.

Network density and Network effectiveness

A SIN with higher density identifies greater interactions and more collaboration among the network agents (providers and customers). In software development, it is evident that effective communication among project members is essential for project success (Suchan and Hayzak, 2001). Specifically, to accomplish the objectives and develop tasks in OSS projects, the effective interchange of information among the development team is extremely essential (Powell, Piccoli et al., 2004). These dense interactions may cause that the information or knowledge to travel through the network in shorter time, however they may affect the time of service delivery. In the networks with higher rank of density, the average number of interactions to fulfill the requested services is more than other networks, and so the service delivery time may increase accordingly over the time. Since in networks with higher density agents may take more time to accomplish the requested services, the number of fulfilled services though a fixed period may decrease accordingly. Thus we propose the hypotheses that:

H1: There is a positive relationship between network density and average service delivery time.
H2: There is a negative relationship between network density and service delivery activity.

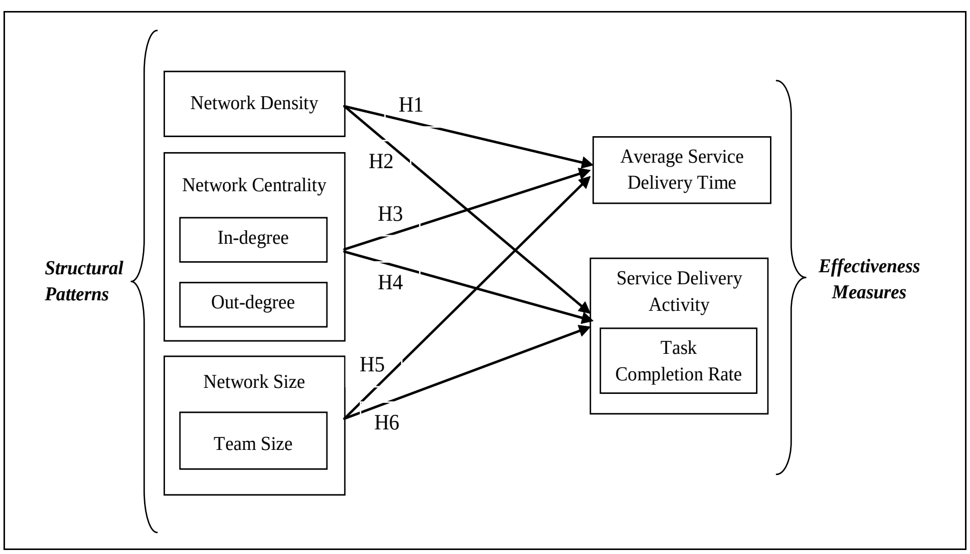


Figure 1. Research Model, SIN Effectiveness Model

Network Centrality and Effectiveness

According to social network theory, the higher centrality indicates the unequal distribution of the interactions in the network. However, it enhances the speed of the information flow within the network (Cummings and Cross, 2003). The agents who are more central, have access to more resources of the network. Since they are generally the most capable agents in the network (Wu et al., 2007), they can filter the unnecessary messages and exchange the information more effectively. The central agents may fulfill the requested service in a shorter time based on their knowledge and experience. On the other hand, most of the requested services need to get the central agents' responds and comments to be executed more effectively. Since the limited time is available for central agents and sometimes they might be overloaded, high score of centrality in SIN may increase the time to accomplish services.

In software development literature, good coordination with better allocation of resources is identified as an important factor for the project success (Kidane and Gloor, 2007). In the SIN with high score in centrality, the role of the central agents is critical to make the service more active. Since they have access to more resources (Wu et al., 2007), their presence and motivation are necessary to promote the level of service delivery activity. As discussed the central agents will be involved to accomplish most of the services, and so the service delivery time may increase accordingly. Clearly, the longer service delivery time may cause the delivery of fewer services over a time period. Thus, the rate of completed tasks in a fixed period decreases, as well as the level of network effectiveness. After all, we suggested the hypotheses that:

H3: There is a positive relationship between SIN centrality and average service delivery time.
H4: There is a negative relationship between SIN centrality and service delivery activity.

Network Size and Effectiveness

The appropriate size for a team depends on the nature of the task (Au et al., 2009). In service delivery systems if a team is too small, it may not accomplish the tasks effectively. For example when the core team members face high workloads, the time to deliver the task may increase, and the quality of their tasks may decrease. Mockus et al. (2002) suggest that the number of core developers in OSS projects may influence the success and performance of the project. Au et al. (2009) provide evidence that suggest that the average bug resolution time is different across various project team sizes. Crowston and Howison (2005) reported that the larger OSS project teams tended to have lower degrees of centrality. In the large teams, more agents are involved to fulfill the tasks and in most cases they are not dependent on the central agents.

Moreover, network size impacts on service delivery activity rate as well. Greater number of core agents in the team can decrease the time for service delivery and the rate of service activity increases accordingly over a time period. Since in the large networks greater number of requests can be assigned to the agents at the same time, the number of completed requests may increase gradually. We hypothesize that the network size has a positive impact on the level of network effectiveness:

H5: There is a negative relationship between network size and average service delivery time.
H6: There is a positive relationship between network size and service delivery activity.

DATA AND METHODOLOGY

The validity and usefulness of the model can only be established through systematic empirical scrutiny To examine the validity the proposed model, we tested the hypotheses using real world data drawn from open source software development projects.

The case of open source software (OSS) development process can be considered as a real world example of SINs. A typical OSS project team delivers its service over the internet. The developers of an OSS project collaborate in different phases of development process (design, architecture , integration, testing and debugging) and collaborate and interact via message boards, email exchanges, etc. (Sawyer, 2004). The communications and interactions among project developers as well as project users form the network of interactions in the project. For instance, a certain bug or feature request is reported via the tracking system of the project and considered as a service request. The developers – service providers – and users – service customers – communicate to fix the bug or to develop the new feature. Their interactions to fulfill the requested service form a network which achieves the outcome of the system (Figure 2). Based on the SIN definition, this setting of OSS project development process can be assumed to be a SIN.

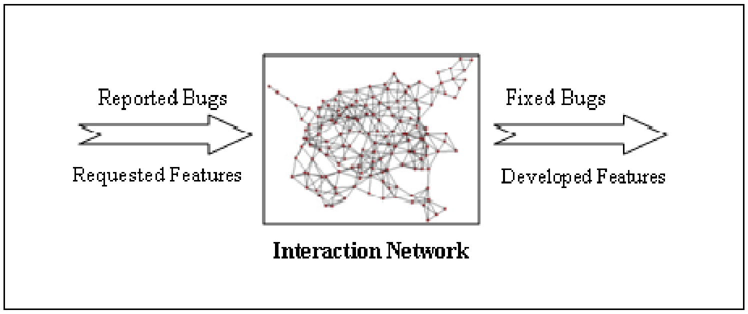


Figure 2. Process Model of SIN in OSS Projects

The data from a sample of OSS projects over one year period (May 2009- April 2010) has been acquired to test the number of specific hypotheses suggested by the proposed model. We used the Multiple Linear Regression (OLS) analysis to test the hypotheses. In this section, the data collection used in this research and the methodology to test the developed hypotheses are described.

Data Sources

The data to analyze the model are acquired from two main sources: 1) SourcForge.net which is the world's largest open source software development and collaboration website, and 2) the SourceForge Research Data Archive (SRDA) (Madey, 2010), which is repository of SourceForge research data made available by the Department of Computer Science and Engineering, the Notre Dame University.

In this study, we used the tracker system in SourceForge for each project which officially tracks and reports project activities such as bug reports, patch submissions, feature and support requests. The tracker system is a tool that allows developers to develop and support software and collaborate with users to fix and manage the reports and submissions (artifacts). Each of those artifacts (like bug reports or feature requests) has basic information, such as artifact ID, priority, submission details, status, assignee details, close date and close time. In addition there are the interaction history between developers and users to fix bugs or develop the features. By studying SourceForge.net website and extract the monthly dumps from SRDA, it is possible to explore developers and users networks structures to fix bugs or develop new features, as well as investigate the impacts of those networks structures on the effectiveness of the system.

Data Collection

A set of projects from various categories on SourceForge.net is selected to validate the model and test the hypotheses. In March 2011 (time for the data collection), SourceForge.net supported over 260,000 open source software projects on a wide variety of topics classified into almost 20 main categories (e.g. Multimedia, Communications, Software development, Education, Security, Database, Social science, Games, Financial, etc.). It is clear that not all of those projects would be suitable for the aim of this study.

According to the SourceForge.net website, some OSS teams are not responding to the reported bugs because they use SourceForge as the 'repository of record' instead of 'repository of use' (Au et al., 2009). Therefore, for those projects there is no interaction network between developers and users to fix bugs or develop new features. In addition, many of them are inactive or just individual projects rather than having sufficient interaction between the project team (Krishnamurthy, 2002). Crowston et al. (2006) examined SourceForge projects over 5-year period (2001-2005) and suggested that almost 67% of the registered projects never had more than one developer at any time of their experiment. Moreover, some of SourceForge projects even do not even have bug reports (tracker tools) availability (Crowston, Howison et al. 2006; Au et al. 2009). In order to collect useful data and analyze the proposed model, we set following criteria to select the appropriate dataset.

- *Project Rank:* The first criterion is the project ranks in SourceForge.net. The internal ranking system in SourceForge uses three sub-factors: traffic, communication and development (Au et al., 2009). By employing this ranking system, the older projects which are dropped in activity have lower ranks. This criterion guarantees that the selected projects are amongst the latest active ones.
- *Project Status:* In SourceForge.net, development status (like inactive, planning, beta, production/stable, mature, etc.) is reported for each project. we selected production/stable projects to ensure that projects are active in terms of contributions to the code repository, request for bug-fixes, supports or features, or in terms of page views (Au et al., 2009).
- *Number of Developers:* The included projects are required to have at least 3 developers. Au et al. (2009) found that the OSS projects with at least 3 developers have lower resolution time (for fixing bugs, developing new feature, etc.) in comparison with 1 or 2 developers. It also may insure that the project is kind of team effort (Crowston, Howison et al.,

2006). Since, this study focuses on the impacts of interactions between developers and users in OSS development; we selected the projects that have at least 3 developers to ensure that they have appropriate number of interactions (Wu et al., 2007).

- *Number of Reported Artifacts:* Since the purpose of this research is analyzing the interaction networks to deliver the requested services, having appropriate number of requests seems necessary. Therefore, the projects which had more than 100 requests (including bug reports, feature requests, support requests and patch submissions) in the project tracker system at the time of selection (March 2011) were selected to the database. This criteria indicates a certain level of development for the project as well (Crowston, Howison et al., 2006).

We selected top 100 projects based on project ranking on SourceForge.net which met the other criteria (Project status, Number of developers and Number of reported artifacts) as well. To analyze the effects of OSS teams' structure on the effectiveness of their network, we monitored each selected project over one year period (May 2009 to April 2010). We captured the interaction records of each project through their tracking systems. We collected data on all types of reports in tracking systems (Bug reports, Feature requests, Patch submissions and Support requests) for each project and collectively call them as Bugs in the rest of this study (Au et al., 2009).

Measures

Based on the proposed effectiveness model on SINs in previous sections, the dependent and independent variables in the model are measures as follows:

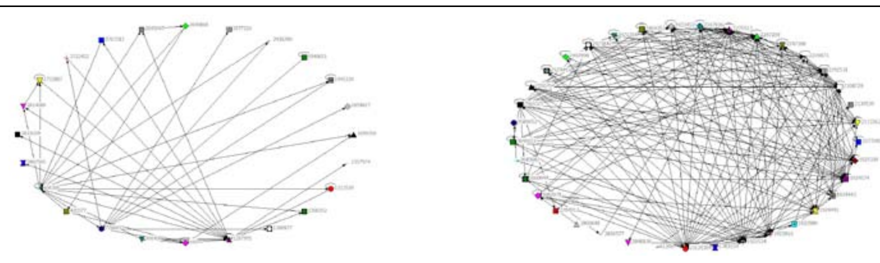
Average service delivery time: In this study, service delivery time is identified by the time spent to fix each bug. It is calculated by opened and closed timestamps recorded in the tracker system. For each project, average of service times for all closed bugs over the one-year period calculated to measure this parameter.

Service delivery activity: It is simply counted by the number of bugs which closed over the period (May 2009 till April 2010) for each project. It indicates the task completion rate (Wu et al., 2007) for each project.

Network density, Network centrality: In order to obtain data on interaction networks, we extracted a *developer/user-artifact* table for each project from the database, including fields such as developers' or users' ID and the artifacts (bugs) in which they are involved and interacted. To make an interaction network from the table and analyze it, Java script applied for each table and the data transformed to an adjacency-matrix for each project. This matrix identifies the interaction network for each project and its cells indicate the number of interactions from one agent (developer or user) to another. To analyze the adjacency-matrix and investigate the score of *network density* and *network centrality* (in-degree and out-degree), we used one of the most popular social network analysis tool of UCInet 6.

Network size: In this study, network size is measured with *team size*. Team size is simply calculated by the number of developers in each project over the period. The number of developers is reported monthly in SRDA data warehouse, so the average of monthly data is calculated as team size for each project over the period.

Examples of interaction networks for two OSS projects from our dataset for one year period (May 2009 to April 2010) are shown in Figure 3 below. The variance in network density, centrality and size between the two projects is striking.



Name: SQUirrel SQL Client

Outdegree centrality: %21.26
Indegree centrality: %18.99
Network density: 0.07
Community size: 24

Name: XCat

Outdegree centrality: %19.74
Indegree centrality: %7.18
Network density: 0.85
Community size: 37

Figure 3. Example of Interaction Network Graphs

ANALYSIS AND RESULTS

We tested all the six hypotheses using OLS multiple regression analysis using Stata/SE11.2 (StataCorp 2009). In this section, first we present the descriptive statistics for the data of 100 OSS projects and our model measures and then the results of examining the impacts of networks structural patterns on SIN effectiveness are presented. Table 1 shows the summary statistics for the acquired data.

According to the calculated mean and median for the reported measures and the examined normality tests, considerable skewness was displayed by the data. To obtain reliable results the log-transformed was used to correct skews. In addition, the correlation summary among the independent variables shows strong correlations between in-degree and out-degree centrality (r=0.86) which may affect the results of the regression models. Therefore, we examined our regression models for those high correlated measures separately which minimize the bias in estimation of the parameters of the model and obtain more reliable results. Table 2 and Table 3 present the results of estimated models based on Equation 1 and 2 respectively. In those models, the network centrality is estimated by its appointed measures separately. In the following sub-sections, we explain how our analysis based on real world data supports the proposed hypotheses of SIN effectiveness model.

Equation 1.

$$ln(SIN\ effectiveness\ parameter_i)= a_0+a_1*ln(Outdegree\ centrality_i)+ a_2*ln(Team\ size_i) + a_3*ln\ (Network\ density_i)+\varepsilon_i$$

Equation 2.

$$ln(SIN\ effectiveness\ parameter_i)= a_0+a_1*ln(Indegree\ centrality_i)+ a_2*ln(Team\ size_i) + a_3*ln\ (Network\ density_i)+\varepsilon_i$$

Characteristics	value
Total number of opened bugs[1]	26903
Total number of closed bugs	22180
Average number of developers for each project	27
Total number of interactions	75328
Average number of responses to each bugs	2.8

Table 1. Overall Summary Statistics for 100 OSS Projects (May 2009-April 2010)

Impacts of Network Density

We tested the hypotheses H1 and H2 which suggested the impacts of network density on SIN effectiveness. The OLS estimates for the effects of network density on service delivery time (measures by average lifespan to fix the bugs) and service delivery activity (measures by the number of closed bugs) are reported in Table 2 and Table 3. H1 proposes a positive relationship between network density and average service delivery time. In line with this hypothesis, our results suggest positive affect of network density on the time to deliver services as well. This positive relationship is the same when network centrality in the estimation models is measured by in-degree or out-degree centrality. H2 suggests negative relationships between network density and service delivery activity. Based on the estimation results of OLS in tables 2 and 3, it seems that there is no relationship between network density and task completion rate which measures the service delivery activity in our model.

Impacts of Network Centrality

The hypotheses H3 and H4 propose the effects of network centrality on SIN effectiveness. We tested these hypotheses and the results for the estimation models are reported in tables 2 and 3. In our model, the network centrality is measured by two attributes: in-degree and out-degree centrality. As discussed because of the high correlation between these two measures, two separated models estimate their impacts on SIN effectiveness.

[1] This value aggregates bugs, feature requests, patches and supported requests

Although H3 suggests a positive relationship between network centrality and average service delivery time, our results do not show a significant relationship between these parameters. However it seems that in the networks with high score of in-degree centrality, the average time to deliver services may decrease. In our model, we proposed negative relationships among network centrality and service delivery activity (H4). The estimation results of OLS regressions in tables 2 and 3 show significant negative relationships between SIN centrality and service delivery activity (which is measured by the number of closed bugs). This result is robust for both network centrality attributes in the proposed model (in-degree and out-degree).

Impacts of Network Size

We also investigated the hypotheses H5 and H6 (the impact of the network size on SIN effectiveness). The OLS regression estimates, reported in tables 2 and 3, show that network size has a significant negative effect on the average time to deliver services. We have proposed a positive relationship between team size and the rate of service delivery activity (H6); however the regression results based on the acquired dataset show no strong relationship between these two measures.

In general, our results lend support to the conjecture of positive effect of network density on service delivery time. Moreover, we have shown that network centrality has a significant negative impact on task completion rate. In addition, our results show that network size can improve the SIN effectiveness in order to decrease service delivery time.

	Out-degree centrality	Team size	Network density
Service delivery time	0.04 (0.80)	-0.25 (0.02)**	0.20 (0.15)
Service delivery activity	-0.91 (0.000)****	0.11 (0.22)	0.06 (0.57)
*P value in parentheses *P<0.1 **P<0.05 ***P<0.01 ****P<0.001*			

Table 2. Results by OLS Regression Model
Equation 1

	In-degree centrality	Team size	Network density
Service delivery time	-0.18 (0.31)	-0.31 (0.008)***	0.30 (0.01)***
Service delivery activity	-1.1 (0.000)****	0.02 (0.69)	0.08 (0.32)
*P value in parentheses *P<0.1 **P<0.05 ***P<0.01 ****P<0.001*			

Table 3. Results by OLS Regression Model
Equation 2

DISCUSSION AND CONCLUSION

The aim of this paper is to examine the effects of the structural patterns of SINs in service delivery on the effectiveness of service systems. Using data from a sample of OSS projects which provide good proxies for SINs in the real world, this study showed that differences in structural patterns of SIN have significant impacts on the effectiveness of the system.

As hypothesized, network density has negative effects on system effectiveness. It is likely that in the networks with high level of density, many redundant and unproductive connections among network agents are possible. Such connections may not necessarily contribute to productive outputs. We also hypothesized that the degree of network centrality has a negative impact on SIN effectiveness. We found that in more centralized networks, the number of completed tasks decreased significantly. It may be interpreted as implying that the core agents are overloaded in more centralized SINs. This can lead to reductions in task completion rates.

Finally, the model proposed that the network size positively impacts the level of perceived network effectiveness. Since the numbers of providers who are involved in large networks are more than smaller groups, average time to deliver services decreases considerably. This is also confirmed by our results. Overall, balancing the structural patterns of the interaction networks (e.g. network centrality, network density and network size) is important to increase the level of service system performance.

The outcomes of this research can be applied in a variety of service organizations that involve extensive communications and interactions between providers and customers to deliver services. The coordinators of service organizations may consider structural patterns of the service delivery networks, and expect higher level of system effectiveness in terms of service delivery time and service delivery activity.

LIMITATIONS AND FUTURE WORK

This study has limitations that suggest future extensions of this research project. In particular, we propose the followings as future research topics based on our work: *First*, one can extend the study by incorporating additional measures of SIN effectiveness (such as service popularity from customer side point of view). *Second*, our model can be tested with datasets from other service sectors (e.g. healthcare, hospitality, etc.) and compare and contrast the results with our findings obtained from OSS development services. This will clarify to what extent our results are driven by the particular service domain from which our data was gathered. *Third*, we limited the structural patterns of our model to network density, network centrality and network size. It will be interesting to investigate the effects of other network structural features such as betweenness centrality on the effectiveness of SINs. *Finally*, besides of the structural patterns of the networks, investigations into non-structural aspects (such as behavioural patterns) as they impact SIN effectiveness have the potential to offer useful insights.

REFERENCES

1. Au, Y. A., D. Carpenter, et al. (2009). "Virtual organizational learning in open source software development projects." Information & Management 46(1): 9-15.
2. Carrington, P. J., J. Scott, et al. (2005). Models and Methods in Social Network Analysis. Cambridge ; New York Cambridge University Press.
3. Coleman, J. S. (1988). "Social Capital in the Creation of Human Capital." The American Journal of Sociology 94(ArticleType: research-article / Issue Title: Supplement: Organizations and Institutions: Sociological and Economic Approaches to the Analysis of Social Structure / Full publication date: 1988 / Copyright © 1988 The University of Chicago Press): S95-S120.
4. Crowston, K., H. Annabi, et al. (2003). Defining open source software project success. Twenty-Fourth International Conference on Information Systems.
5. Crowston, K. and J. Howison (2005). "The social structure of free and open source software development." First Monday 10(2-7).
6. Crowston, K., J. Howison, et al. (2006). "Information Systems Success in Free and Open Source Software Development: Theory and Measures." Software process improvement and practice 11: 123–148
7. Crowston, K. and B. Scozzi (2002). "Open source software projects as virtual organisations: competency rallying for software development." Software, IEE Proceedings - 149(1): 3-17.
8. Cummings, J. N. and R. Cross (2003). "Structural properties of work groups and their consequences for performance." Social Networks 25(3): 197-210.
9. Dixit, K., S. Kameshwaran, et al. (2009a). "Service Interaction Networks: Simulation and Applications." IBM Research Report.
10. Dixit, K., S. Kameshwaran, et al. (2009b). "Towards simultaneously exploiting structure and outcomes in interaction networks for node ranking." IBM Research Report.
11. Ewusi-Mensah, K. (1997). "Critical issues in abandoned information systems development projects." Commun. ACM 40(9): 74-80.
12. Freeman, L. C. (1978). "Centrality in social networks conceptual clarification." Social Networks 1(3): 215-239.
13. Gunasekaran, A., C. Patel, et al. (2001). "Performance measures and metrics in a supply chain environment." International Journal of Operations & Production Management 21(1/2): 71-87.
14. Hinds, D. and R. M. Lee (2008). Social Network Structure as a Critical Success Condition for Virtual Communities. Hawaii International Conference on System Sciences, Proceedings of the 41st Annual.
15. Jennings, E. T., Jr. and J. A. G. Ewalt (1998). "Interorganizational Coordination, Administrative Consolidation, and Policy Performance." Public Administration Review 58(5): 417-428.
16. Kameshwaran, S., S. Mehta, et al. (2009). Analyses for service interaction networks with application to service delivery. Proceeding of SIAM International Conference on Data Mining.

17. Kameshwaran, S., V. Pandit, et al. (2010). Outcome Aware Ranking in Interaction Networks. CIKM-10. Toronto, Canada.
18. Kidane, Y. and P. Gloor (2007). "Correlating temporal communication patterns of the Eclipse open source community with performance and creativity." Computational & Mathematical Organization Theory 13(1): 17-27.
19. Krishnamurthy, S. (2002). "Cave or Community?: An Empirical Examination of 100 Mature Open Source Projects." First Monday 7(6).
20. Madey, G. The SourceForge Research Data Archive (SRDA). University of Notre Dame. (Last accessed: October 2010).
21. Maglio, P. and J. Spohrer (2008). "Fundamentals of service science." Journal of the Academy of Marketing Science 36(1): 18-20.
22. Maglio, P. P., S. Srinivasan, et al. (2006). "Service systems, service scientists, SSME, and innovation." Commun. ACM 49(7): 81-85.
23. Milward, H. B. and K. Provan (2003). "Managing the hollow state, Collaboration and contracting." Public Management Review 5(1): 1 - 18.
24. Mockus, A., R. T. Fielding, et al. (2002). "Two case studies of open source software development: Apache and Mozilla." ACM Trans. Softw. Eng. Methodol. 11(3): 309-346.
25. Page, S. (2003). "Entrepreneurial strategies for managing interagency collaboration." Journal of Public Administration Research and Theory 13(3): 311(329).
26. Powell, A., G. Piccoli, et al. (2004). "Virtual teams: a review of current literature and directions for future research." SIGMIS Database 35(1): 6-36.
27. Provan, K. G. and H. B. Milward (1995). "A Preliminary Theory of Interorganizational Network Effectiveness: A Comparative Study of Four Community Mental Health Systems." Administrative Science Quarterly 40(1): 1-33.
28. Provan, K. G. and H. B. Milward (2001). "Do Networks Really Work? A Framework for Evaluating Public-Sector Organizational Networks." Public Administration Review 61(4): 414-423.
29. Sawyer, S. (2004). "Software development teams." Commun. ACM 47(12): 95-99.
30. Scacchi, W. (2002). "Understanding the requirements for developing open source software systems." Software, IEE Proceedings - 149(1): 24-39.
31. Scott, J. (2000). Social network analysis: A handbook. London, Sage.
32. Scott, J. (2002). Social Networks: Critical Concepts in Sociology. London, New York, Routledge.
33. StataCorp (2009). "Stata Statistical Software: Release 11." College Station, TX: StataCorp LP.
34. Suchan, J. and G. Hayzak (2001). "The communication characteristics of virtual teams: a case study." Professional Communication, IEEE Transactions on 44(3): 174-186.
35. Turrini, A., D. Cristofoli, et al. (2010). "Networking Literature about Determinants of Network Effectiveness." Public Administration 88(2): 528-550.
36. Wasserman, S. and K. Faust (1994). Social Network Analysis: Methods and Applications. Cambridge, Cambridge University Press.
37. Wu, J., K.-Y. Goh, et al. (2007). Investigating Success of Open Source Software Projects: A Social Network Perspective. ICIS 2007, Montreal, Canada.