Information Brokers in Requirement-Dependency Social Networks

Sabrina Marczak, Daniela Damian, Ulrike Stege, and Adrian Schröter

Department of Computer Science - University of Victoria
3800 Finnerty Road, Victoria, British Columbia, Canada
{smarczak, danielad, stege, schadr}@cs.uvic.ca

Abstract

Requirements interdependencies create technical dependencies among project members that generally belong to different functional groups in an organization, but who need to coordinate activities during processes of requirements change management. Effective knowledge management is needed to disseminate information on requirement changes across teams working on interdependent requirements to avoid misinterpretations. Social networks are regarded as important in fostering knowledge management, where brokers or gatekeepers have the role of project members facilitating information flow. However, little is known about processes of information flow and brokerage in social networks built around interdependent requirements.

In a field study of requirement interdependencies in a large IT manufacturing organization, we found that brokers holding pockets of knowledge have an impact on information flow in requirement-interdependent teams. We discuss a number of patterns of information flow and draw implications for processes of requirements change management.

1. Introduction

Recently, we have seen a growing interest in the study of collaboration in software and requirements engineering [2,11,13]. While the match between coordination needs created by technical dependencies and social structures has been studied in the context of socio-technical congruence [13] and awareness [2,11], we take a requirements-oriented point of view. In our work we study aspects of requirements-driven collaboration, which we relate to processes of communication and coordination in cross-functional teams that work on the same or interrelated requirements. We study broader coordination that takes place across diverse functional teams by considering technical dependencies as created by requirements interdependencies.

As the concept of requirements-driven collaboration is largely unexplored, details about communication structures that facilitate effective knowledge flow across dependent, cross-functional teams are not only important but also missing in literature. Our research goal is to explore the information flow patterns found in teams that are coordinating work on interdependent requirements. We report our findings on a specific type of information flow, namely brokerage of information between members of interdependent teams.

Figure 1 is an illustration of brokerage as an information flow pattern between the social networks of Team 1 and Team 2 working on requirements $R_1$ and $R_2$ respectively. Given a pair of requirements $R_1$ and $R_2$ where $R_2$ depends on $R_1$, we are interested in those information flow patterns between the teams allocated to work on these requirements. Specifically, we investigate the presence of particular team members that channel the outgoing information from Team 1 to Team 2 (e.g., $p_4$ in Figure 1), the incoming information to Team 2 from Team 1 (e.g., $p_5$ in Figure 1), and the consulting flow between members of an outside team (e.g., $p_5$ for pair $p_4, p_3$). These team members are examples of brokers.
Brokers and their roles have been studied in organizational behavior as an important aspect of information flow. Yang and Tang [17] found that teams with fewer brokers and more directed edges reported fewer performance problems. Hauschild and Schewe [14] indicate the important role of gatekeepers (brokers) in propagating innovation throughout an organization. In requirements-driven collaboration brokers emerge as important roles because their presence has implications to processes of requirements change management. To use the example in Figure 1, brokers such as p4 and p5 are in a position to mediate information flow between members who would otherwise have no direct connection (e.g., the pairs p1, p3 and p4, p5). Although brokers can act as useful mediators between groups that are geographically distributed (e.g., [10]) or in teams where formal communication structures enforce communication through certain central people in the organization, usually technical leaders [16], there are also potentially negative consequences when they mediate requirements information. Not only is a broker in a position to control the transfer of information about changes from members in Team 1 to those whose work is affected in Team 2, a broker may also introduce noise in the information being carried.

Our research questions thus pertain to the presence and properties of such brokers in requirements-driven collaboration, and broader patterns of information flow within which these brokers emerge and operate.

The remainder of this paper is organized as follows: Section 2 discusses the background and related work. Section 3 introduces the research questions, and Section 4 describes our research methodology. Data analysis and discussion of the results are presented in Sections 5 and 6, respectively. Section 7 outlines threats to validity, before we conclude in Section 8.

2. Background and Related Work

In our recent work [1], we defined the concept of a requirement-centric social network (RCSN) as a social network that represents a cross-functional team where each member is involved in a particular aspect of a requirement’s development (e.g., design, code or test). Each connection between the members in an RCSN represents a relationship such as communication with, or awareness of, each other. A project member may belong to one or more RCSNs and a project has as many RCSNs as unique requirements. By relating the team members who work on the same requirements we can study how people collaborate, maintain awareness of work, and coordinate based on requirements-related tasks. Dependencies among members of an RCSN represent technical dependencies and create coordination needs that transcend the team boundaries typical for activities of requirements engineering, design, implementation, or testing.

In this paper we explore the collaboration among members in multiple RCSNs. We define a requirement-dependency social network (RDSN) as a representation of two or more requirement dependent RCSNs. Figure 2 illustrates a situation where Requirement R1 depends on Requirement R4. The resulting RDSN includes those project members that work on both requirements.

![Figure 2. Illustration of the RCSN and RDSN concepts applied to a pair of interdependent requirements](image)

Dependency relationships between requirements can be of structural nature (e.g., refined-to, changes-to and similar-to dependencies), constraining nature (e.g., requires, and conflict-with dependencies) or cost/value nature (e.g., increases/decreases cost of dependencies) [3]. Changes to a requirement in any of these dependency relationships need to be propagated to those working on the other requirements. Thus, project members in what we call “interdependent networks” must have coordination needs. If a requirement changes, one would expect that effective requirements change management and expertise seeking processes across these networks would make use of adequate communication, awareness and information flow processes. In our research we make a first step in studying communication structures and patterns of information flow within interdependent networks and draw implications for requirements change management and coordination in software development.

2.1 Knowledge Flow and Group Communication Structure

In the study of teams, social networks are recognized as important in fostering relationships, trust and knowledge management [4,5]. This is because, as literature on small groups indicates, group members dif-
fer in their expertise, knowledge, and information they bring to the group [6]. Further, group performance depends not only on such information resources being available through the groups, but also on the processes and structures used to utilize these resources in the social networks [7].

It has been argued that the distribution of knowledge among members of a group affects the group performance. Wherever information is broadly distributed across group members, these members share common information and conceptualization, and this in turn facilitates group performance. In groups where pockets of unique knowledge are concentrated in specialists, critical information possessed by certain members may never be shared or retrieved by the group, and this can diminish group performance [5]. However, one factor that may modify the effect of distribution of knowledge is the group structure, conceptualized as the quality and patterns of working or social relationships among group members.

Studies show that the group structure can either constrain or facilitate knowledge and information flow within a group. Somewhat centralized information flow, for example through brokers, would represent a hierarchical communication structure. Similarly, flat networks or decentralized groups provide opportunities for task-oriented communication and information exchange [8,9]. Such a dense communication structure enables the free flow of information and thus broader distribution of knowledge. The performance in groups of generalists with broadly distributed knowledge does not seem to be affected by the group structure. Whereas centralized groups of specialists were outperformed by decentralized groups [5].

On the other hand, group structure seems to play an important role when one considers knowledge dissemination across teams. Past studies of information flow in distributed groups (e.g., [10]) found that distributed teams with more centralized structures experienced fewer coordination problems, whereas dense communication was associated with more coordination problems. Some hierarchy caused by particular team members acting as point people (i.e., brokers) through whom much of the cross-team communication flowed, was identified as critical to ensuring coordination across teams [10].

2.2 Information Flow and Brokers in Requirement-Dependency Social Networks

In studying collaboration in interdependent networks, we posit that members in RCSNs are specialists. Given each project member’s different role in the project, a member of the cross-functional team has specific knowledge about his respective requirement. For example, the requirements analyst may have different knowledge about the requirement than the developer or tester. The study of communication within RCSNs should reveal insights about communication patterns that may modify the effect of uneven knowledge distribution on the team’s performance. Similarly, studying information flow and group communication structures in RDSNs may reveal insights about the dynamics of collaboration and knowledge brokerage among people that need to coordinate their requirements-related work.

In order to investigate information flow in interdependent requirements networks, we analyzed field data from an industrial software project. We examined the project requirements and planning documentation and studied the information flow in pairs of requirement interdependencies. This is a first step in our long-term investigation about requirements-driven collaboration.

We call a requirement that is dependent on another requirement dependent and denote it $R_i$. The requirement that $R_i$ depends on is called dependee and is denoted $R_j$. We denote such a pair of dependent requirements $R_iR_j$ to be read as $R_i$ depends on $R_j$. The information flow from the network $R_iCSN$ to the network $R_jCSN$ of dependent $R_j$ is the information that is carried from members belonging to $R_iCSN$ to members belonging to $R_jCSN$. This information can be carried through any number of distinct members or be propagated directly.

For an idealized communication between $R_iCSN$ and $R_jCSN$, information from each member in $R_iCSN$ should be transferrable to all members in $R_jCSN$. The longer the path used to transfer information, the higher the chance for misinterpretation and loss of information. Therefore, short paths of information-travel are of special interest in requirements management processes, namely direct communication between members of $R_iCSN$ and $R_jCSN$ as well as communication along paths of length two, which involves a single mediator, referred to as a broker, in between members in $R_iCSN$ and $R_jCSN$ respectively.

3. Research Questions

First, we turn our attention towards studying information flow brokers within RDSNs. We believe that not only the existence of brokers is important but also that different types of brokers facilitate different kinds of information flow. For example, when a requirement changes, one would expect that information about the change would be propagated to those project members that work on every dependent requirement. Brokers who act as key people in sending the information from
the dependee requirement network to those in the dependent network are in a position to transfer this information between project members who otherwise may have no direct communication path (e.g., $p_4$ in Figure 1). This situation is similar to brokers that mediate the information sent into the dependent network from those in the dependee network (e.g., $p_5$ in Figure 1). Further, members that mediate flow between members of a different team take a consulting role and may serve as sources of expertise not available in the other team (e.g., $p_3$ for pair $p_4, p_3$ in Figure 1). Thus our first research question is:

**RQ 1.** Are there different types of brokers in interdependent requirements networks?

More specifically, we ask.

**RQ 1.1.** Are there brokers in the dependee network $RCSN$ who mediate information to the dependent network $RCSN$?

**RQ 1.2.** Are there brokers in the dependent requirement $R_i$ who mediate information from the dependee network $RCSN$ to $R_i$?

**RQ 1.3.** Are there consulting flows for members in any of the $R_iCSN$ or $RCSN$?

Certain brokers may also play a crucial role in information flow processes beyond directly mediating communication between two members. In general, people are considered crucial if they disrupt the communication across the interdependent networks, when removed. For example, we see in Figure 1 that if we remove $p_4$ then most members of Team 1 cannot reach the members in Team 2. Hence, $p_4$ has a large influence on the communication between teams. Therefore, we ask the second research question:

**RQ 2.** Are there members who are crucial to the information flow in RDSNs and if so, what are their roles?

### 4. Research Methodology

We conducted an exploratory case study in the Brazilian software development center of a large international IT manufacturing company, which we refer to as ORG for confidentiality reasons.

#### 4.1 Research Setting

ORG assembles and ships its products worldwide, and has an extensive IT department to support its internal processes. The project we studied updates and maintains an internal software product used by ORG to support its shipping process. The product is eight years old, and is a critical component of the company’s business. As the product is critical to the company, delivery of the project on time is the main measure of the project success. The project was delivered on time and all requirements were met.

The project duration was five months. Our observations were during a three-month period, from the beginning of the Development phase to the third week of the Test Execution phase. The project team was distributed between the USA ($n=4$) and Brazil ($n=10$), as follows: the project manager, the system architect, one development leader, and a senior developer were located in USA, and a development leader, five developers (including one senior developer), a test leader, and a three testers were located in Brazil. In addition, there are business partners, located in the USA, who define the project requirements. Although the business partners are internal clients, they are not members of the project team. Thus, the business partners were not included in our study.

The project members’ experience with ORG ranged between less than one year and more than seven years, with an average of two and a half years. Both developer leaders have the most knowledge about the product and are most familiar with all team members. Two project members declined participation, the project manager and the system architect. Project guidelines recommend that developers and testers are not supposed to contact business partners unless through one of the leaders. Thus, business partners communicate primarily with the project manager and the developer leaders.

#### 4.2 Modeling the Information Flow

Drawing on the terms of dependent requirements and information flow, we say for a particular communicating pair of members belonging to the pair of dependent requirements $R_i R_j$, where $R_i$ depends on $R_j$, that information is sent by a member denoted $s$ (sender) and received by member $r$ (receiver). If there does not exist direct communication from $s$ to $r$, then we call a person $b$ who is directly receiving information from $s$ and sending it directly to $r$, a broker. We say that the information from $s$ to $r$ is brokered by $b$. We are interested in the following three types of brokers or information flow:

**Outgoing Information Flow** is the topic of interest in research question RQ1.1. We say that the brokered information flow with respect to $(s, r)$ leaves the dependent network if $s$ belongs to the dependee requirement network, $r$ belongs to the dependent requirement network, and a broker $b$ for $(s, r)$ belongs to the dependee requirement network. In short, we call this flow outgoing information flow with respect to $(s, b, r)$. See brokers $b_1$, $b_2$, $b_3$, $b_5$, $b_7$, and $b_8$ in Figure 3 for the configurations of this flow.
Incoming Information Flow is the topic of interest in research question RQ1.2. We say a brokered information flow with respect to \((s,r)\) enters the dependee requirement if \(s\) belongs to the dependee requirement, \(r\) belongs to the dependent requirement, and a broker \(b\) for \((s,r)\) belongs to the dependent requirement as well. In short, we call this flow incoming information flow with respect to \((s,b,r)\). Brokers \(b_5, b_6, b_9, b_{10}\), and \(b_9\) in Figure 3 relate to the configurations for this flow.

Additionally, we are interested in communication between members of the same network that are mediated by an outsider to that requirement network. We call such a broker also consultant.

Consulting Information Flow is the topic of interest in our research question RQ1.3. A broker \(b\) for \((s,r)\) indicates a consulting flow if \(s\) and \(r\) belong to \(R_jCSN\) and \(b\) belongs to \(R_iCSN\), or \(s\) and \(r\) belong to \(R_iCSN\) and \(b\) belongs to \(R_jCSN\). Brokers \(b_1, b_9,\) and \(b_{10}\) in Figure 3 shows possible configurations. Observe that configuration \(b_{10}\) can be mirrored.

We extended the brokerage concept proposed by Gould and Fernandez [12] in order to include the definition of brokers in the intersection area of interdependent requirement networks and to consider the indication of information flow. Having described the types of brokers and information flow, we describe next the data collection methods as well as our data model.

### 4.3 Data Collection

To construct requirement-dependency social networks, we collected data through document inspection, interviews with and questionnaires to project members, as well as follow-up interviews to collect missing data and to validate the already collected data.

From the inspection of project documentation, we identified a total of eighteen requirements. To collect data for each requirement, by inspecting the project plan we first identified a list of people who are allocated to work in every task related to the particular requirement. To build the communication ties in the social networks, we deployed a questionnaire in the third month of the project’s life cycle, at the beginning of the Test Execution phase. Using the questionnaire we collected data on who talked to whom about which requirement.

The questionnaire was customized for each respondent. Each project member was asked to indicate whether he had communicated with a respective person in a provided list. This list of members was created in advance, and included those project members working on the same requirements as the respondent.

To obtain details about the communication, we asked the participant to indicate the type of communication in which he was engaged. We provided a list of five types of communication, which we identified as relevant to the project through interviews earlier in the study: Communication of Changes, Coordination of Activities, Requirement Clarification, Requirement Negotiation, and Synchronization of Code.

To identify sets of interrelated requirements, we examined the requirements-traceability document and conducted interviews to validate the type and our understanding of these dependencies. We identified five pairs of interdependent requirements. Using the terms introduced in Section 2, out of the five pairs, two are structural dependencies and three are constraining dependencies. Furthermore, we also interviewed project members about requirements management practices, and work and communication structures.

### 4.4 Data Model

For each pair of dependent requirements \(R_i, R_j\) and each type of interaction \(K\), we constructed a separate RDSN. Thus, we have twenty-five RDSN (five pairs times five types). Such a social network \(SN(R_i R_j K)\) is represented by a graph with nodes (actors) representing team members associated with the two requirements and directed ties indicating the type of communication \(K\) between two actors. More precisely, \(SN(R_i R_j K) = (A_i, A_j, E)\), with \(E \subseteq A_i \times A_j\), where the set of actors \(A_i\) models people belonging to dependee requirement \(R_iCSN\), and the set of actors \(A_j\) models people belonging to dependent network \(R_jCSN\). For every tie \(a_i a_j \in E\) we assume that \(a_i\) communicates with \(a_j\) directly under \(K\). Note that the set of actors who are members of both \(A_i\) and \(A_j\) is not necessarily empty,
that is \( A_i \cap A_j \neq \emptyset \) is possible. For convenience, we denote with \( A_i \backslash A_j \) the set of actors in \( A_i \) who are not also in \( A_j \). Vice versa, \( A_j \backslash A_i \) denotes the actors in \( A_j \) who are not members of \( A_i \).

To construct RDSNs, we first create the two RCSNs from questionnaire information about who talks with whom about each type of communication regarding a certain requirement. For each requirement \( R \), when a participant reported that he communicated with someone about certain type of interaction \( K \), we created a tie in the respective requirement network with the type of interaction between the respondent and the person he indicated. By repeating this process for all respondents we built the RCSN for each requirement and type of interaction (step b in Figure 2).

For each pair of interdependent requirements \( RR \) and each type of communication, we join the respective RCSNs and generate the requirement dependent social network \( SN(R,RR,K) \), see Figure 2, step (c). For a pair of members whose communication was reported in RCSN but not in RCSN (or vice versa), we assumed the communication took place in the RDSN. Thus, a tie between the members is indicated in the corresponding RDSN. Note that members belonging to both RCSN and RCSN are then located in the overlapping area of the RDSN.

We remark that the two members who declined participation in our study were not involved in the interdependent requirements investigated. Thus, the missing data of those members did not impact our analysis and results.

The broker roles investigated in this paper are modeled as follows. Given a network composed of actors and directed ties, brokerage occurs if, in a triad of actors \((s,b,r)\), \(sb,br \in E \) and \( sr \notin E \). I.e., \( s \) has a tie to \( b \), \( b \) has a tie to \( r \), but \( s \) has no tie to \( r \). We call \( b \) a broker for \( s \) and \( r \). A broker \( b \) for the pair of actors \((s,r)\) represents the outgoing flow in \( SN(A_i,A_j,K) \) if \( s \in A_j \), \( r \in A_i \), and \( b \in A_j \). A broker \( b \) for the pair of actors \((s,r)\) represents the incoming flow in \( SN(A_i,A_j,K) \) if \( s \in A_j \), \( r \in A_i \), and \( b \in A_i \). A broker \( b \) for the pair of actors \((s,r)\) is called a consultant for \( A_i \) and \( A_j \) in \( SN(A_i,A_j,K) \) if either \( s,r \in A_i \), or \( b \in A_j \backslash A_i \), or \( s,r \in A_j \), and \( b \in A_i \backslash A_j \).

5. Analysis and Results

In this section we describe how we analyzed the data, largely in relation to each of our research questions. For readability, we include the findings for each of the research question immediately after describing the data analysis methods.

To identify brokers and distinguish the different types of brokers in our RDSNs, we implemented an algorithm based on the brokerage concept defined by Gould and Fernandez [12]. We computed all relevant brokers in the twenty-five RDSNs and identified a number of brokers for each network. Table 1 shows all forty-six brokers we identified and which correspond to configurations in Figure 3. Each network, represented by a cell in Table 1, can have up to ten broker types and each person can take the role of each broker. For each pair we also indicate the median of people in each RDSN as membership. For example, in Table 1, Pair 1 (1,6,1) indicates that there is one actor allocated in RCSN, six belonging to both RCSNs, and one to RCSN. We observe that the brokers concentrate in the first three types of interactions, namely Communication of Changes, Coordination of Activities and Requirements Clarifications. Each broker in each network is identified by the configuration and person (e.g., \( b_5:p_1 \)). Some people are associated with multiple configurations.

RQ1.1. Are there brokers in the dependee network RCSN who mediate information to the dependent network RCSN?

To answer this question, we identify those configurations in Table 1 that refer to broker of outgoing flow \((b_1, b_2, b_3, b_4, b_5, \) and \( b_6 \) in Figure 3). Out of the total of 46 instances of brokerage, 37 indicate outgoing flow. They are distributed over 17 of the 25 RCSNs as follows: \( \#(b_1) = 0; \#(b_2) = 4; \#(b_3) = 4; \#(b_4) = 16; \#(b_5) = 0; \) and \( \#(b_6) = 13 \). These results indicate that most of the outgoing brokers are in the overlapping areas (i.e., \( A_i \cap A_j \)), meaning that they are project members that work on both interdependent requirements.

The brokers for outgoing flow are enumerated below, together with their role in the project, geographical location and number of occurrences over the total number of outgoing brokers. One particular person \((P_3)\) emerges most frequently across all networks. This person is a USA-based Dev Leader. The distribution of all brokers is as follows: \( P_1 \) (Test Lead, BR): 8/37; \( P_2 \) (Dev Lead, BR): 6/37; \( P_3 \) (Dev Lead, USA): 22/37; \( P_4 \) (Tester, BR): 1/37; and \( P_5 \) (Senior Dev, USA): 0/37.

RQ1.2. Are there brokers in the dependent requirement RCSN who mediate information from the dependee requirement RCSN to RCSN?

To answer this question, we identify those configurations in Table 1 that refer to broker configurations of incoming flow \((b_2, b_3, b_4, b_5, b_6, b_7, \) and \( b_8 \) in Figure 3). Out of the 46 cases, 42 indicate incoming flow. These brokers were identified in 16 out of the 25 RCSNs, and are distributed as follows: \( \#(b_2) = 4; \#(b_3) = 3; \#(b_4) = 16; \#(b_5) = 5; \#(b_6) = 1; \) \#(b_7) = 13; \#(b_8) = 1. Similar to the outgoing flow, these results indicate that most of the incoming brokers are project members that work on both interdependent requirements.
A similar trend can be observed by examining the distribution of people in these configurations. Same USA-based Dev leader (P3) is the most frequent broker. The distribution of all brokers is: P1 (Test Leader, BR): 8/42; P2 (Dev Leader, BR): 6/42; P3 (Dev Leader, USA) 18/42; P4 (Tester, BR): 1/42; P5 (Senior Dev, USA): 7/42; and P6 (Senior Dev, BR): 2/42.

**RQ 1.3. Are there consulting flows for members in any of the R_CSN or R_V_CSN?**

To answer this question, we identify those configurations in Table 1 that refer to broker configurations of consultant flow (b1, b9, and b10 in Figure 3). Out of the 46 cases, only 5 indicate consultant flow. They took place in 4 out of the 25 networks, and are distributed as follows: #(b1) = 4; #(b9) = 1; #b10 = 0. The brokers for consulting flow belong in 4 out of 5 cases to both requirements.

The distribution of brokers for the consulting flow is as follows: P3 (Dev Leader, USA): 4/5 and P2 (Senior Dev, USA): 1/5. Again, P3 is the dominant person. P3 and P2 are in USA and are very active with Brazil. P3 mediated twice between Brazilian members and twice between Brazil and USA; P2 mediated between members located in Brazil.

**RQ2. Are there members who are crucial to the information flow in RDSNs and what are their roles?**

We identified the necessity of the members in each RDSN for maintaining the information flow. To do this, we determined every actor b in the network whose removal disconnects an existing information flow for a pair of actors (s,r).

In total, thirty-six crucial actors were identified across all networks (seven distinct members). The actors identified in bold in Table 1 are crucial actors. The distribution of crucial people is as follows: P1 (Test Leader, BR): 11; P2 (Dev Lead, BR): 3; P3 (Dev Lead, USA): 12; P4 (Tester, BR): 2; P5 (Senior Dev, USA): 4; P6 (Tester, BR): 3; and P6 (Dev, BR): 1.

The two most frequent crucial actors are P3, the USA Dev Lead, and P1, the Test Lead, who is located in Brazil. All crucial actors are also reported brokers, except P7 and P8. It is interesting to observe, however, that the most dominant broker (P3) is also the most important crucial actor.

### Table 1. Distribution of brokers cases per configuration and person

<table>
<thead>
<tr>
<th>Pair</th>
<th>Communication of Changes</th>
<th>Coordination of Activities</th>
<th>Requirements Clarification</th>
<th>Requirements Negotiation</th>
<th>Synchronization of Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,6,1</td>
<td>b1: P1</td>
<td>b2: P2</td>
<td>b3: P3</td>
<td></td>
<td>b4: P3</td>
</tr>
<tr>
<td>1,6,4</td>
<td>b1: P1</td>
<td>b2: P1, P3</td>
<td>b3: P3</td>
<td></td>
<td>b4: P3</td>
</tr>
<tr>
<td>1,5,4</td>
<td>b1: P5</td>
<td>b2: P1</td>
<td>b3: P5</td>
<td></td>
<td>b4: P3</td>
</tr>
<tr>
<td>0,6,0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>1,8,2</td>
<td>b1: P1, P3</td>
<td>b2: P3</td>
<td>b3: P3</td>
<td></td>
<td>b4: P3</td>
</tr>
</tbody>
</table>

After analyzing the number of brokers in each of the five pairs of requirements, for each communication type in part, we now discuss patterns of brokerage across the twenty-five networks we studied.

The presence of brokers in any of these networks indicates the existence of pairs across two interdependent requirement networks that do not communicate directly and for which the information flow was mediated. In the context of requirements engineering, this indicates that those members whose communication was brokered did not use and possible had no direct way of communicating requirements information. Misunderstandings or information loss could occur in these mediated interactions. We clarify that information on the flow of a unit of information from sender to broker and then from broker to receiver were not collected. Instead, we analyzed the flow of information in general between members, without knowing if the same specific unit of information was transferred. We believe that our discussion of implications for requirements
engineering is the best that can be drawn from an understanding of information flows without tracking specific messages.

6.1 Brokerage Predominant in Certain Types of Communication

Brokers of information flow were identified in eighteen out of the twenty-five RDSNs we investigated. A pattern that emerged is that the majority is concentrated in the first three requirement pairs and is almost evenly distributed over only three types of communication: Communication of Changes, Coordination of Activities and Requirements Clarification.

When these findings are considered in the context of requirements engineering, they can be explained by the nature of interaction inherent in these communication types. Most project members are involved in communicating changes, coordinating activities, and clarifying requirements on a daily basis. In contrast, the activities of Requirements Negotiations or Synchronization of Code may be less frequent and also only involve certain members. ORG’s guidelines require that team leaders negotiate requirements with business partners, and thus requirements negotiations should happen outside the development team and brokerage happens with the business liaison. Similarly, Code Synchronization activities have significant tool support and are mostly automated. Most communication is due to synchronization problems. Given that the project we studied was successful, we conclude that brokerage was acceptable for these types of communication and not detrimental to the team performance.

Further, the seven networks without brokers are not equally distributed across Table 1 but mostly located with Pair 4. This finding can be explained by the fact that all team members in Pair 4 are part of both teams and we studied only brokerage in which at least one member is in only one requirement network.

6.2 Does Distance Really Matter?

Perhaps the most interesting and surprising pattern is that the most frequently identified broker in all configurations, and also a crucial member in all but one pair of requirements, is located in the USA (P3, fictitious name Jane). One would not be surprised that Jane mostly communicated across distances given that ten out of fourteen project members are in Brazil, and because she played a key role in the project, as a development leader. Given that geographical distribution introduces significant communication problems [13], and that maintaining relationships across distances is known to be difficult [15], one would expect that for more efficient communication she would have appointed or collaborated with a Brazilian-based project member or leader.

Surprisingly, that was not the case. Jane was not only communicating actively with the distanced members but was a broker for all communication types among Brazilians. We would have expected to see this pattern with a Brazilian-based project member. Particularly interesting is that she was a consultant in two cases, despite the distance from Brazilian members. This can be explained by the familiarity of members with Jane given her role, and corroborates with prior evidence that familiarity has a positive influence on communication patterns [15]. Other factors that could explain this effect are knowledge and experience in the project, as discussed next.

This finding implies that organizations planning to establish a remote team with requirements-driven technical dependencies could mitigate the effect of distance by ensuring that the remote team includes experienced team members.

6.3 Knowledge and Experience as Determinants for Brokerage

Additional contextual project information reveals other factors that relate to brokerage. Allocated to three out of eighteen requirements, Jane was part of four of the five pairs of interdependent requirements. We believe that her knowledge and experience is a strong determinant for her broker role in most networks. Jane has been a development leader at ORG for more than seven years. She not only acquired extensive knowledge of the project in her role as coordinator of negotiation activities with the business partners, but ORG also has the practice of copying development leaders on all project-related email communication. Her profile fits what has been referred to as a specialist role and leads her to become a broker in the team’s communication. Her ability to act as a broker of communication with the distanced members is an example of a group structure in which active communication counteracts the effect of knowledge pockets in organizations.

6.4 Brokers as Gatekeepers of Information

When particular configurations of brokerage were considered, special types of brokers emerged that mediated the flow between interdependent networks. This corroborates with the literature on the role of gatekeepers in coordination processes of interdependent work teams [11,14].

In the context of requirements engineering, brokers who act as information flow agents between two
interdependent requirements networks emerge as gatekeepers in processes of requirements change management. For example, a broker from $R_i$CSN that mediates outgoing information to $R_j$CSN is in a position to control the flow between those who need to send the change information to those who need to receive it because their work is affected by the change and need to effectively coordinate. Similarly, brokers from $R_i$CSN that mediate incoming information into $R_j$CSN will be in a position to control the receiving of change information and thus affect the ways in which those in $R_j$CSN can react to change.

Two interesting patterns of outgoing and incoming brokerage emerged. First, as expected, most brokers of information flow are those members who are related to both requirements. On the one hand these people have most knowledge about people and their work in both networks. In the context of requirements management processes, these brokers would be in a better position to know who in $R_i$CSN needs to receive the change information when sent from $R_j$CSN. On the other hand, in our case study most project members in the RDSN were related to both requirements. This results in a higher chance that brokers are located in the intersection of two requirement networks.

Second, one would expect that most frequent configurations that needed brokerage would be those in which the sender and receiver of information work in different requirements and the broker would be located at the intersection of the two networks. However, the most frequently observed brokerage was for the communication between a member in only one requirement and a member working on both requirements (e.g., brokers $b_3$ and $b_4$ in Figure 3). This is surprising, given that in all these cases the members’ work assignment to the same requirements would imply direct communication not mediated through brokers.

7. Threats to Validity

We faced three distinct threats that deal with the team structure, the examined communication path, and the restriction to certain flows.

Team Distribution. As to be expected, the two teams working for a pair of dependent requirements $R_i,R_j$ did share a substantial number of team members. The median of people in the intersection $A_i \cap A_j$ for each pair is as follows: Pair 1 has 6 of 8, Pair 2 has 6 of 11, Pair 3 has 5 of 10, Pair 4 has 6 of 6, and Pair 5 has 8 of 11 people in the intersection. It explains the high occurrences of particular brokers within the intersection $A_i \cap A_j$ of the two teams. We could not anticipate that prior to the study.

Communication Paths. In our investigation we assumed that a single message from a sender $s$ can be communicated to a receiver $r$, as long as a directed path from $s$ to $r$ exists in $SN(R_i,R_j,K)$. However, this may not be the case, as in our data collection we only collected information about communication ties between pairs of people, but not the exact message they were communicating about. Therefore, a direct link may allow the sending of information but does not necessarily imply that a certain message was sent. However, we did collect information about the specific topic of communication (e.g., communication of changes) and this indicates that channels that we constructed over those pairs were most likely to convey the desired information type.

Flow Completeness. Our case study does not provide a complete picture of all the possible information flows. While we did report the complete information flow in $SN(R_i,R_j,K)$ from actors in $A_i \setminus A_j$ to $A_i$ (including the overlap $A_i \cap A_j$) and from $A_i$ (including the overlap $A_i \cap A_j$) to $A_j \setminus A_i$, we did not investigate configurations where both sender and receiver were in the overlap $A_i \cap A_j$, as we had no information whether the communication was (1) about one of the requirements only, (2) going from members in $R_i$CSN to $R_j$CSN respectively. Although we collected data on all flow directions, we reasoned in the beginning of this study that not all flows are relevant in our analysis.

8. Conclusion

Requirement misinterpretation is a well-known problem in requirements engineering and is more likely to occur when information is not propagated effectively. However, people who act as mediators do exist in organizations because of formal organizational structures such as informal practices that consider experience, knowledge, or simply geographical distribution [10, 13].

In this study we investigated the presence of brokers and ways in which brokers influence information flow. We have analyzed patterns of information flow in pairs of interdependent requirements in a successful project of a large IT manufacturing company.

Our findings indicate that there are key members, namely brokers, controlling the information flow from the dependee network $R_i$CSN to the dependent network $R_j$CSN. Furthermore, we found that the power of information flow lies within the hands of only a few members.

The most compelling insight was the presence of one information flow agent that played the central role across all interdependent pairs. Additionally, she was not even physically collocated with the majority of the
team. We posit that her extensive knowledge and experience in the organization, as well as familiarity with the project members, allowed her to overcome the challenges of distance and effectively provide brokerage for the remote project members.

Still, our study only scratched the surface of the relationship between social networks, information flow and requirements change management. In this study, we investigated pairs of interdependent requirements, the simplest case of dependency among requirements. We plan to explore the scalability of our methodology by analyzing data from a large project with more requirements involved in a dependent relationship. Moreover, our analysis represents only a snapshot of communication patterns in the project. We understand that social interaction and networks develop over time in organizations, thus we intend to further investigate the evolution of these patterns over time.

We also plan to study the relationship between information flow pattern in requirement-dependency networks and more specific project performance measures such as coordination ease. It is our endeavor to understand which patterns of information flow facilitate greater socio-technical congruence in requirements-driven collaboration.

Acknowledgements

Special thanks to the ORG project team members, who volunteered in our study. We appreciate the insights provided by our colleagues Irwan Kwan, Thanh Nguyen-Huynh, Lucas Panjer, and Timo Wolf. We also thank NSERC and University of Victoria for financial support.

References