On the role of boundary spanners as team coordination mechanisms in organizationally distributed projects

Anh Nguyen-Duc
Norwegian University of Science and Technology
Trondheim, Norway
anhn@idi.ntnu.no

Daniela S. Cruzes
Norwegian University of Science and Technology
Trondheim, Norway
dcruzes@idi.ntnu.no

Reidar Conradi
Norwegian University of Science and Technology
Trondheim, Norway
reidar.conradi@idi.ntnu.no

Abstract—Software projects are still facing challenges of team coordination across global boundaries. Boundary spanner is an important organic coordination mechanism that is not much explored in GSD literature. This paper presents a finding from four case studies of how a boundary spanner resolve coordination gaps in organizationally distributed teams. The qualitative data were collected from 16 interviews from different types of global software projects. Boundary spanners have common characteristics of a coordinator, such as team member recognition, multiple perspective expertise, decision-making ability and work time flexibility. Task negotiation, conflict resolution, task information navigation and boundary object set-up are common activities to support team coordination. We also discussed a compound effect of other organizational roles and the impact of context factors on boundary spanner’s activities.

Keywords—global software development, case study, boundary spanning, team coordination, organizational boundary

I. INTRODUCTION

Recognized as a popular phenomenon in software industry, global software development (GSD) has received a significant interest from both researchers and practitioners in last decades. Although many collaborative development infrastructures [1, 2, 13-15], processes [3, 11, 12] and practices [4, 5, 7, 10] have been proposed to facilitate global software projects, GSD continues to face many coordination challenges across global boundaries [6]. Many proposed solutions to manage coordination problems in GSD are assumed to work in a homogeneous environment without considering functional variations in engineering process, various work infrastructure, tools and development practices, different objectives and strategies among organizational units [25]. While establishment of homogeneous development and collaborative environment is difficult, connecting team members and transferring necessary information in a timely manner across organizational boundary is essential for an effective coordination.

One of the mechanisms to address these boundary issues is to rely on individuals who obtain the role of linking the organization’s internal networks with external sources of knowledge and information. Boundary spanners are considered as a standard organizational design element in product development organizations. Formally nominated or informally emerged, boundary spanners are intended to facilitate knowledge sharing, transporting information, translating requirements and transferring expertise [17]. Boundary spanning activities are found as a significantly influenced factor on team collaboration across boundaries and a predictor of several different organizational outcomes such as knowledge sharing, team coordination and team effectiveness [18].

In SE context, the role of boundary spanner in coordinating team activities is not much explored. Boundary spanner is investigated under different terms, such as “liaison”, “knowledge mediators”, “information broker”, “bridge engineer” [19-23]. Boundary spanner role can be taken by a requirement analyst, a project manager or a special engineer to cross boundaries between customers and development teams. Depending on the organizational role it takes, a boundary spanner might have different abilities and perform different kind of coordination activities in different context. Although there are some studies investigating coordination activities and practices as a team effort [16], there is no systematic investigation of this role from an individual viewpoint.

Prior to this study, we revealed a research gap on how team coordination can impact performance of GSD teams [25]. Then, we identified a set of coordination challenges in inter-organizational context [16], i.e. coexistence of competition and collaboration attitude [24]. As a next step, this study explores a specific type of coordination, so-called boundary spanners. Firstly, to understand boundary spanning as a coordination mechanism, we would like to identify characteristics of a boundary spanner that made them a perceivedly effective coordinator [26, 27]. These characteristics include abilities, skills and expertise that a boundary spanner possesses and support them in coordinating technical tasks.

RQ1. What characteristics of boundary spanners contribute to effective coordination in distributed software projects?

Secondly, we would like to understand what has been done by a boundary spanner to support team coordination. Once a boundary spanner possesses necessary capabilities, it is important to know how they enable effective coordination.
RQ2. In which way boundary spanners handle dependencies in distributed software projects?

The remainder of the paper is structured as follows: Section 2 presents the background and related work. Section 3 describes research method and Section 4 presents our findings. Section 5 provides a discussion and Section 6 concludes the papers.

II. BACKGROUND

A. Software dependencies and team coordination

There are many definitions of team coordination in organization science, information management and other disciplines. In this study, we understood team coordination as “activities required to maintain consistency within a work product or to manage dependencies within the workflow” [28]. There are several types of dependencies among software development tasks, such as: task dependencies (i.e., needed to integrate software parts seamlessly), temporal dependencies (i.e., necessary to synchronize activities and adhere to project schedules), software process dependencies (i.e., required to adhere to the established software processes and act as agreed with other team members), and knowledge dependencies (i.e., a team needed business domain knowledge from the other team to accomplish a task) [28, 30]. Failure to support these dependencies results in a coordination breakdown, for example, lack of team awareness and mutual trust, limited choice of coordination mechanism, difficulty in task scheduling, delay in communication and lack of common understanding [29].

Several frameworks of coordination activities have been proposed, such as organic coordination, and mechanistic coordination [31]. Organic coordination (mutual adjustment, or social coordination mechanism) is the use of lateral communication means to coordinate activities [31]. Mechanistic coordination is the use of vertical communication means to coordinate activities in a programmed way, such as direct supervision and standardization, task organization, role assignment, schedules, plans, division of labor, project controls and specifications, routine meeting and status check [31]. A boundary spanning activity can be classified as either organic or mechanistic and might be influenced by the role of boundary spanners in the organization.

B. Coordination across organizational boundary

Coordinating information and knowledge across boundaries is a constant activity in production management due to the appearance of communities of practices [32]. Compared to a geographically distributed but organizationally unit project, an inter-organizational team needs to handle not only geographical, temporal, cultural boundaries, but also the difference in organizational objectives, engineering process, and team collaboration attitudes [8, 24, 46]. Grechanik et al. showed the coordination gap between developers and testers in GSD [8]. Gopal et al. revealed different goals and expectation between vendor and client’s organization in software outsourcing industry [46]. Anh et al. showed that competition and collaboration could coexist in inter-organizational software projects [24]. In this study, we have both cases illustrated vendor-to-vendor relationship and vendor-to-client relationship. Paasivaara et al. revealed a set of coordination practices across organizational boundaries, such as synchronization of main milestones, frequent deliveries, establishment of peer-to-peer links and relationship building [16]. However, none of these practices are investigated from a boundary spanner’s viewpoint.

C. Boundary spanner roles and boundary spanning activities

Early work viewed the boundary spanner as a role linking “organizational structure to environmental elements, whether by buffering, moderating, or influencing environments” [33]. Boundary spanners are stakeholders who translate and frame information from one environment to another in an effort to promote coordination [33, 34]. Boundary spanner roles can be occupied by different team members, such as project managers, technical leaders, requirement analysts and external coordinators [19-23]. Tushman et al. found that characteristics of boundary spanners was specialized to meet the information requirements of both their projects and their information boundary [54]. The boundary spanners might not be the source of new information and ideas but they have to gather and disseminate them to other team members [55]. Di Marco et al. stated that boundary spanners were not necessarily formal team leaders or project managers, but can be any of the team members who connect the members of distributed sub teams in the project network [22]. Levina et al. studied offshore collaboration in information systems development and found that a middle manager on the onshore team could act as a boundary spanner to mediate the negative effects arising from status and cultural differences [35].

Boundary spanning activity can be performed in team level to establish linkages and to manage interactions with parties in external environment [36-38]. Boundary spanning activities have a purpose of meeting collective performance goals and capturing the interactions across the team to parties in embedding environment such as clients, customers, industry experts, and other mutually interdependent teams. Examples of boundary’s activities include representing the team to outsiders (e.g., updating and seeking feedback from upper management on team progress), searching for information (e.g., contacting subject matter experts for project-related expertise), and coordinating task performance with other external groups (e.g., communicating action plans and delivery deadlines with other interdependent teams [37]). Recent work in IS revealed that boundary spanners-in-practice must actually engage in boundary spanning by relating practices of one team to practices of another team and negotiating the meaning and terms of the relationship [17].

In addition to boundary spanner, boundary objects can be used to coordinated team member’s activities as well. Boundary object is described as an artifact that “sits in the middle” of diverse knowledge groups, establishing a “shared and sharable” context for distributed problem solving. These artifacts need to be “both plastic enough to adapt to local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites”
In IT systems, boundary objects can be document archives, source code, design artifacts, physical prototypes, design documents and repository [35, 40].

D. Boundary spanner roles and boundary spanning activities

Carmel et al. found an ability of boundary spanner to cross cultural boundary by resolving cultural gaps and translating linguistic information [41, 42]. Pawlowski et al. described an ability to facilitate sharing of expertise, integrating communication of the team [39]. Faraj et al. highlighted an ability to protect inner team and obtain political support [43]. In our context, we defined boundary spanning activities as a mechanism that (1) brings team members into contact with non-members or (2) overcomes knowledge and information boundary by engaging agents from different knowledge communities in collective activities. Furthermore, we looked at effective activities by in-practice boundary spanners.

Strode et al. looked at boundary spanning as a coordination strategy in four co-located projects [44]. Compared to their conceptual framework, we showed that boundary spanning not only supported explicit coordination (right place, right thing, right time [44]) but also facilitated the exchange of know-who, know-what and know-how information in GSD teams. Besides, we highlighted a boundary leveraging function of boundary spanners in distributed context, which were not visible from collocated projects [44]. Krishnan et al. examined the usefulness of boundary spanning in collaborating distributed teams via project social capital perspective [45]. Compared to their conceptual model (main constructs are spanning knowledge boundary, social boundary and operational boundary), we showed that spanning practice boundary took a role in influencing team coordination. While Krishnan’s model based on literature, our framework is grounded from empirical data in four different types of global software projects.

III. RESEARCH METHODOLOGY

A. Case study design

1) Case selection

We performed exploratory case studies in different global project context [47]. The unit of analysis for each individual case was a GSD project, each project involved more than one organizations. Due to agreements about confidential information with interviewee participants, we used aliases instead of real project identity, so-called Project River, Project Mountain, Project Ocean and Project Tree. On one hand, to represent different global boundaries, we selected Project River, Project Mountain and Project Ocean that represents for different global arrangement setting, such as insourcing, offshore outsourcing and inshore outsourcing [48]. To compare the interaction between boundary objects and boundary spanner, we selected Project Tree as an extreme case since it is inherently different from the other three projects (mainly relied on set of boundary objects). On the other hand, to be able to reason about influences of different global setting on team coordination, we selected cases with similar set of coordination mechanisms and practices. All four cases had some common coordination mechanisms, such as configuration management system, email exchange, site visiting, and social relationship building.

Certain practical consideration also influenced case selection. Project River, Mountain and Tree were selected from a pool of companies located in our location. In Project Ocean, data was collected conveniently from our researcher’s professional networking. The selected projects needed to last long enough to ensure its maturity and also enough data for conducting the research. The selected projects needed to be currently active so the stakeholders could recall and describe current events. Besides, we maximized variety across cases to provide better concept definition by considering a variety of contexts and factors in the emergent theoretical concepts [49]. Projects were selected with different product types, product domain, team size and team organization. Details of these four cases are summarized in Table 1-2.

2) Data collection

We relied on data triangulation as defined by Yin [50]. Our data collection based on: (1) interviews as a primary data source, (2) documentation, and passive observations as a secondary data sources. We conducted 16 interviews with different project roles, such as project manager, technical leader and developers. The interviews were recorded and then transcribed. Two observation sections in Project River and one online observation section in Project Tree were performed. Overall, we collected 65 transcript A4 pages and some other documents, such as meeting minutes, emails and observation notes as material for analysis. All transcripts were sent to interviewees for confirmation. The interview guide was semi-structured and consisted of mainly open-ended questions, hence allowing follow-up discussion, or change of question order. The interview duration varied from 45 to 75 minutes. The interview guide consisted of five parts: (1) information about projects, interviewees and their roles in the project, (2) team coordination challenges created by organizational boundaries, (3) perception of collaboration infrastructure and tools as a coordination mechanism, (4) perception of boundary spanner and boundary spanning activities as a coordination mechanism, (5) reflection on an overall effectiveness of team coordination. It is noticed that the interview questions varied slightly among cases due to the applicability of questions and availability of information. The complete set of questions is given in Appendix A.

3) Data analysis

We analyzed the collected data in two-stage analysis, following the recommendations by Miles and Huberman [51]. In first stage, we conducted a with-in case analysis, started by reading from each case all interview transcripts and relevant documents. The same data analysis framework was used for each case. The interesting text that expresses either challenges with organizational boundary, boundary spanner activity, boundary object and boundary leveraging practices were labeled with an appropriate code. Any text that links these concepts were coded as well.
Table 1: Boundary spanner identification

<table>
<thead>
<tr>
<th>Case</th>
<th>Team role</th>
<th>Spanner type</th>
<th>Reason for being selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>River</td>
<td>Team leader</td>
<td>Nominated</td>
<td>Team leader of Team Thames was responsible for communicating with Team Danube and customers.</td>
</tr>
<tr>
<td>Mountain</td>
<td>Program manager</td>
<td>Nominated</td>
<td>Program manager of Team Dom worked closely with other teams, formed an informal integration group across locations.</td>
</tr>
<tr>
<td>Ocean</td>
<td>Bridge engineer</td>
<td>Nominated</td>
<td>Bridge engineer in team Baltic traveled to the onsite site and work closely with customer team. He was responsible for communicating and delivering tasks to offsite teams.</td>
</tr>
<tr>
<td>Tree</td>
<td>Gatekeeper</td>
<td>Emerge</td>
<td>Gatekeeper was responsible for giving patches, finding and fixing bugs and participating in communication with other project participants.</td>
</tr>
</tbody>
</table>

After that, we grouped relevant codes in to higher-order code using axial coding [47]. Transcript data was entered into the qualitative data analysis tool NVivo 8 for the ease of qualitative coding. At this point, no comparison between cases was made. In second stage, the concepts and their relationship were compared among four cases to identify commonalities and also variations among them, and to find a list of good practices [50].

B. Boundary spanner identification

We adopted several ways to identify an in-practice boundary spanner. Boundary spanner was known as an individual involved in communication with external stakeholders, i.e. testing team, geographically distributed teams, customers, third-party and acknowledged by at least a team member about his connecting role. For each project, we selected one boundary spanner as a representative sample. Boundary spanners were selected by:

- Identifying team members of the projects and construct team structure (All cases)
- Asking references from team members (All cases)
- Confirmation by the boundary spanner (All cases)
- Social network analysis (Project Tree)

The description of selected boundary spanners is shown in Table 1. In Project Tree, a social network analysis on communication data in mailing list and issues tracking systems clarified the role of these gatekeepers [9]. In the selected time frame (10 years) for this study, 224 of the 342 firm-paid developers in Project Tree were active in the mailing list, the issue tracking system or both. These 224 developers represented for 132 different commercial firms. We identified top 20 developers with highest betweenness and closeness for social network based on mailing list and bug tracking system. One of them was accepted for interviewing and he also confirm about his communication with other firms in the projects.

C. Case description

Table 2 describes characteristics of each project and data collected from them.

1) Project River

Context: Project River developed and maintained a ship management system. The project was more than five years old and included a large amount of legacy code. The project involved two external contractors, so-called Team Thames and Team Danube. Team Thames located in Trondheim, Norway and implemented the web application system. Team Danube located in Stavanger, Norway and implemented the desktop application system. The development methodology was Scrum. At the beginning of the project, there was little dependency between two teams as both teams only shared a common data access layer and database. At the time this study was conducted, both teams were performing a joint effort on refactoring the overlap code base, updating data model and developing a new module. Communication between two teams and with customer teams was performed mainly via phone and email. Site visits and workshop occurred sometimes during the project duration.

Coordination effectiveness: communication across team was mainly done via the boundary spanner. There was a common adoption of collaboration technology but it was not effectively used to coordinate developer tasks due to the misalignment of governance policy and lack of commitment on using the tools. Therefore, most of the time, developers from both sites did not know about other team’s status. Boundary object such as desktop screen, paper prototype, code guideline was mainly used to support the cross-boundary activity of the team leader and other developers. Boundary spanner was considered as an effective mechanism in this case.

2) Project Mountain

Context: Project Mountain developed a search engine system as part of a large IS management system. The search engine system had many API interfaces, which were used by many higher-level systems. The project was more than five years old, involving more than 1500 developers in a global scale. The part of the project that we investigated only related to a team in the US headquarters and two development centers in Norway, namely Team Everest, Team Dom and Team Eiger, correspondingly. Therefore, there were high amount of technical, temporal and process dependency among these teams. Intense team coordination was performed in daily basis. Team Everest developed a part of search functions of a large IS system, while Team Dom and Team Eiger were responsible for developing core modules of this search engine. The common set of coordination infrastructure and process was governed among all teams, including daily (virtual) meeting, teleconferencing meeting, email and TFS. Team
Eiger and Dom had some local collaboration practices, such as communication via GIT, informal chat and frequent site visit.

**Coordination effectiveness:** a group of program managers acted as a coordinator board to facilitate implementation of cross-team tasks. A set of collaboration infrastructure, such as collaborative tools, teleconferencing system was adopted in all locations. Therefore, it made an impression that developers got a necessary support to understand a big picture of who-do-what. The main coordination issue is to cope with time zone difference and limited face-to-face meeting. The program managers were also perceived as an effective coordinator.

3) **Project Ocean**

**Context:** Project Ocean developed a sub-system of a large-scale e-government solution. The project was more than six years old and its subsystem was outsourced to a development team in Vietnam during the last three years. The project involved a client team in Japan and several vendor teams in Asia Pacific area. In the scope of this work, considered only team coordination among the client team and a vendor team in Vietnam, namely Team Baltic and Team Bering. Team Bering adopted Agile approach while Team Baltic used a tailored water fall development process. The tasks were sent to Team Bering iteratively and integrated to the main system by a joint team. Coordination between two teams were done via email, site visit and bridge engineer.

**Coordination effectiveness:** Boundary spanner was also the primary collaboration channel. Due to the large difference on practices between vendor and customer organization, the collaboration tool did not give much help. Team Behring relied heavily on the bridge engineer as more than 50% of his task devotes for team communication. The uncertainty of tasks from customer increased the passive activities of boundary spanner. The act of boundary spanners helped to reduce risks for knowledge management process to extract and maintain tacit knowledge of the projects in the organization, and threats of information bottom neck in the social network.

4) **Project Tree**

**Context:** Project Tree developed a network package analyzer. The project was open source, on-going, ten-years-old and attracted a lot of participation from commercial firms. The development and product roadmap was done by a small group of active volunteer and commercial developers. Project main communication channels were mailing list and issue tracking system. The project also organized annual conference to exchange experiences and practices among developers and users of the product. We studied the coordination between a commercial firm Y and the Project Tree community. Firm Y developed a prototype and integrated modules from Project Tree, there was a need to understand technical insight and also keep up-to-date with changes in the project.

**Coordination effectiveness:** In Project Tree, there was a heavily rely on tools and collaborative objects to substitute for other coordination mechanisms. As participants were assumed to have the same interest when participating in the project, the variety in knowledge and practices was reduced. The self-explained software patch and issue description form enabled coordination without communication.

<table>
<thead>
<tr>
<th>Table 2: Case description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project</strong></td>
</tr>
<tr>
<td><strong>Project type</strong></td>
</tr>
<tr>
<td><strong>Global arrangement</strong></td>
</tr>
<tr>
<td><strong>Project age</strong></td>
</tr>
<tr>
<td><strong>Product</strong></td>
</tr>
<tr>
<td><strong># intw.</strong></td>
</tr>
<tr>
<td><strong># obsr.</strong></td>
</tr>
<tr>
<td><strong>Interv interw role</strong></td>
</tr>
</tbody>
</table>

As a result, there was little problem regards to team coordination in Project Tree. The boundary spanner in this case was considered as a reliable source of information for team members.

**IV. RESULTS**

A. **RQ1:** What characteristics of boundary spanners contribute to effective coordination in distributed software projects?

The in-practice boundary spanner capacity expresses an ability to attract team member recognition, holding different type of expertise, decision making ability and work time flexibility, as shown in Table 3.

**Team recognition:** in all cases, boundary spanners gain trust from colleagues in both the collocated location and remote locations. From a team player’s perspective, the boundary spanners were seen as a closest position to external stakeholders and have influence on them: “I believe that the project owner trust us deeply. That is almost the problem. They trust us so much that they consider [Team leader name] as a part of the internal technical team. If [Team Thames] explodes and disappears the customer will have problem. I don’t think [Team Thames] is easy to be replaced at this time” (a developer from Team Thames, Project River). Boundary spanners were recognized by not only internal team members, but also external stakeholders as an interface to communicate with the other project stakeholders. For a customer, boundary spanners were considered as a source of information and also a target for giving feedback and negotiating tasks (Project Mountain, Project Ocean).

**Multiple area expertise:** in project River, the team leader had knowledge of not only client side systems but also server side business logic and implementation framework: “[Team leader name] has worked with customer for so long, so he knows about technical detail and domain logic much more
than we do” (a developer from Team Thames, Project River).
In project Ocean, the bridge engineer hold knowledge about
customer requirement, development framework and also
expertise in testing: “One capacity of a bridge engineer is to
be keen on the technology and the development framework,
but also be fast on learning business domain and requirement
from customers ... As part of my task, I also checked
deliverables from [Team Behring] before submitting into
customer main branch.” (a developer from Team Behring,
Project Ocean). In Project Tree, a gatekeeper is keen on a
general network protocol as well as the specific domain
protocol that is locally used in Firm Y. The gatekeeper also
works on several projects that use the network protocols.

Decision making ability: an important capacity of
boundary spanner is the ability to make decisions. In Project
River and Mountain, decision on how to leverage boundary
distances and to assign tasks were parts of spanner’s
responsibility: “But when you run the projects and try to
resolve the conflict and dependencies, it is all happening up
here (management level). A lot of project tracking is done up
here. For a big project if this one is late what consequences to
the other team? I need to involve in all these stuffs” (a project
manager from Team Eiger, Project Mountain).
Especially, in Project Mountain, the project manager can make
decisions on whether to recruit contractors from other
locations. In Project Ocean, a bridge engineer was nominated
certain rights to make decisions in the onsite locations:
“Sometimes, there are many small issues that you need to
decide onsite, it will be unnecessary and too much
bureaucracy for asking permission. So normally we are
assumed to be a decision maker for these issues” (a developer
from Team Behring, Project Ocean).

Work time and place flexibility: as a special position that
stands across multiple boundaries, including temporal and
cultural boundary, a spanner needs to be flexible in working
time and place.

Table 3: Boundary spanning capacities

<table>
<thead>
<tr>
<th>Concept</th>
<th>Code</th>
<th>R</th>
<th>M</th>
<th>O</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team recognition</td>
<td>Internal team’s recognition</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>External team’s recognition</td>
<td>o</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Multiple area expertise</td>
<td>Business logic and technical expertise</td>
<td>o</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Inter-features technical expertise</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Inter-project technical expertise</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Decision making ability</td>
<td>Task assignment</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Team configuration</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td></td>
<td>Management decisions</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Work time and place flexibility</td>
<td>Ability to travel</td>
<td>x</td>
<td>x</td>
<td>o</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ability to work over time</td>
<td>x</td>
<td>x</td>
<td>o</td>
<td></td>
</tr>
</tbody>
</table>

x represents for the presence, while o represents for the absence of the code, R(River), M(Mountain), O(Ocean), T (Tree)

In Project Mountain, program manager often need to shift
work time or work overtime to participate in meeting from
other time zone: “... last night I had 4 meetings in a row from
9h30 am to midnight. We did it from home and all necessary
tools were available... Office is empty at 5 pm. People go
home, have dinner and if we need to do some works we do it
later...” (a project manager from Team Eiger, Project
Mountain). In Project Ocean, a bridge engineer needs to stay 6
months in the customer sites for the first time and has frequent
visits after that: “A bridge engineer needs to be ready for a
frequent onsite trips. Last year I stayed 6 months in the [Team
Baltic]. This year, I have visited them 3 times, but in a shorter
trips”. In Project Tree, the work time on the OSS project was
mixed of office work time and personal time: “Initially, I
participated in the [Project Tree] as my personal interest. As
the company hired me, I started to promote [Project Tree] to
the company. Nowadays, I work on the features needed by the
company in my official work time. At home, I work on other
things” (a gatekeeper from Project Tree).

B. RQ2. In which way boundary spanners handle
dependencies in distributed software projects?

Boundary spanners perform four types of activities that
resolve team interdependencies, which are mediating task
dependency, mediating status information, leveraging global
boundaries and mediating practice flow, as shown in Table 4.

Mediating task dependency is the most frequently
mentioned concept, which includes four sub-concepts, namely
comprehending and transferring tasks, negotiating tasks and
integrating deliverables. There was often a discussion thread
between a spanner and customers or other team (via formal
meetings or exchanged emails) to clarify a new task
linguistically and semantically (Project Ocean, Project River,
Project Mountain). In Project Mountain, the program manager
wrote requirement description of a new task and communicate
request of changes from other teams. We also observed that
boundary spanners also take care of dividing tasks for other
team members: “I know nothing about them [Team Danube,
Project River]. Actually I receive tasks from [the team leader
name, Team Thames, Project River], he is not exactly a
project manager but more like a project organizer. So he
received the tasks from [Danube team] and split the tasks for
us ...” (a developer from Team Thames, Project River).

Moreover, the scope and schedule of a task were
negotiated by boundary spanners between vendor teams
(Project River) or between a customer team and a vendor team
(Project Ocean, Project Mountain). Negotiating often required
a prior experience and knowledge about similar tasks: “I need
to estimate a feasible duration for completing the task. This is
often done based on my previous experience. If the customer
give too short period for a tasks, I need to report to the project
manager and negotiate this if necessary” (a bridge engineer
from Team Bering, Project Ocean). In Project Tree, the
gatekeeper had done some discussion to influence the
development of a feature to be aligned with his company’s
requirement.
In Project River, Ocean and Tree, boundary spanners also involved in integrating and validating source codes from team members before delivering to customers or other vendor teams. A developer from Team Thames said: “[A new task] will be coordinated with [Team Danube, Project River]. The project leader is doing some merging today and after he is done with those, I will merge my changes to the main branch and he will merge all to the development branch”. The integration was done by allowing checking into boundary spanner’s development branch (Project River) or by sending patches (Project Ocean, Project Tree) or by sending emails with the copy to relevant product owners and vice versa. For some tasks, I am informed about [Team Danube] from the customer” (a team leader from Project River).

Interestingly, boundary spanners not only transport the status information but also filter some information to be shared. The information, such as detail development process (Project River, Project Ocean) and specific technical insights (Project Tree) were considered by boundary spanners when giving information about status of a task: “Normally the customer wants to know everything about what happens in the process. As we have some customized processes, we do not share all of information about how we develop the functionality. I was told to make decisions about negotiating these types of issues as well” (a bridge engineer from Team Bering, Project Ocean).

**Leveraging global boundaries**, such as geographical boundary, temporal boundary and cultural boundary is also a part of boundary spanner’s activities. We observed that site visit was commonly adopted in Project River, Mountain and Ocean to reduce negatives effect of geographical boundary. Shifting work time or even going onshore was done in a project with small overlap time window. In case Ocean, a bridge engineer was also a translator of requirement documents for local teams: “...one of my task is to translate requirement documents into a friendly and localized document for the development team at home...” (a bridge engineer from Team Bering, Project Ocean).

**Mediating practice flow** is the least mentioned concept from data transcript. In Project Mountain, the gatekeeper facilitated the exchange of development practices and sharing knowledge expertise as part of organizational learning process: “We have a policy that people should visit [Team Everest] once per year. The purpose is to get to know people, to make the communication smoother for distance collaboration. If you have met a person in the same room and done some work with him it is much easier to pick up the phone later. And also to understand the other site how they are work. Seating far away you may ask why they are doing that? You don’t know how the team is organized” (a program manager from Team Eiger, Project Mountain).

**Mediating status information** consists of three sub-concepts: liaising, transporting information and hiding information. The bridge engineer in Project Mountain and the program manager in Project Ocean communicated project status as one of their main tasks. Information about who do what, and current status of development tasks was formally exchanged in a daily meeting of the manager group (Project River) or weekly report to customer team (Project Ocean). In Project Tree, a gatekeeper subscribed for changes from the OSS community and informed management level about the status of the release.

In Project River, the team leader was the only person in Team Thames was informed about Team Danube’s work status from the customer. This information was not frequently circulated in Team Thames, but when a problem arises, such as a bug or possible delay of a joint task, Team Thames members were informed by the team leader: “I need to be sure that everyone got what they needs, nothing more. We often have indirect contact with [Team Danube] via customers. We almost always send emails with the copy to relevant product owners and vice versa. For some tasks, I am informed about [Team Danube] from the customer” (a team leader from Project River).

### Table 4: Boundary spanning practices

<table>
<thead>
<tr>
<th>Concept</th>
<th>Code</th>
<th>R</th>
<th>M</th>
<th>O</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mediating task dependency</td>
<td>Comprehending and transferring tasks</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>o</td>
</tr>
<tr>
<td></td>
<td>Integrating deliverables</td>
<td>x</td>
<td>o</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Negotiating tasks</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Resolution of task conflicts</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Mediating status information</td>
<td>Liaising</td>
<td>x</td>
<td>x</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td></td>
<td>Transporting status information</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Hiding information</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Leveraging global boundaries</td>
<td>Translating linguistic information</td>
<td>o</td>
<td>o</td>
<td>x</td>
<td>o</td>
</tr>
<tr>
<td></td>
<td>Shifting work time</td>
<td>o</td>
<td>x</td>
<td>x</td>
<td>o</td>
</tr>
<tr>
<td></td>
<td>Site visit</td>
<td>x</td>
<td>x</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Mediating practice flow</td>
<td>Facilitating exchanged practices</td>
<td>x</td>
<td>o</td>
<td>o</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Sharing knowledge and expertise</td>
<td>o</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Setting up boundary object</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Enforce practices around boundary object</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>x</td>
</tr>
</tbody>
</table>

x represents for the presence, while o represents for the absence of the code, R(River), M(Mountain), O(Ocean), T(Tree)

In Project River, Ocean and Tree, boundary spanners also involved in integrating and validating source codes from team members before delivering to customers or other vendor teams. A developer from Team Thames said: “[A new task] will be coordinated with [Team Danube, Project River]. The project leader is doing some merging today and after he is done with those, I will merge my changes to the main branch and he will merge all to the development branch”. The integration was done by allowing checking into boundary spanner’s development branch (Project River) or by sending patches (Project Ocean, Project Tree) or by sending emails with the copy to relevant product owners and vice versa. For some tasks, I am informed about [Team Danube] from the customer” (a team leader from Project River).

Interestingly, boundary spanners not only transport the status information but also filter some information to be shared. The information, such as detail development process (Project River, Project Ocean) and specific technical insights (Project Tree) were considered by boundary spanners when giving information about status of a task: “Normally the customer wants to know everything about what happens in the process. As we have some customized processes, we do not share all of information about how we develop the functionality. I was told to make decisions about negotiating these types of issues as well” (a bridge engineer from Team Bering, Project Ocean).

**Leveraging global boundaries**, such as geographical boundary, temporal boundary and cultural boundary is also a part of boundary spanner’s activities. We observed that site visit was commonly adopted in Project River, Mountain and Ocean to reduce negatives effect of geographical boundary. Shifting work time or even going onshore was done in a project with small overlap time window. In case Ocean, a bridge engineer was also a translator of requirement documents for local teams: “...one of my task is to translate requirement documents into a friendly and localized document for the development team at home...” (a bridge engineer from Team Bering, Project Ocean).

**Mediating practice flow** is the least mentioned concept from data transcript. In Project Mountain, the gatekeeper facilitated the exchange of development practices and sharing knowledge expertise as part of organizational learning process: “We have a policy that people should visit [Team Everest] once per year. The purpose is to get to know people, to make the communication smoother for distance collaboration. If you have met a person in the same room and done some work with him it is much easier to pick up the phone later. And also to understand the other site how they are work. Seating far away you may ask why they are doing that? You don’t know how the team is organized” (a program manager from Team Eiger, Project Mountain).

**Mediating status information** consists of three sub-concepts: liaising, transporting information and hiding information. The bridge engineer in Project Mountain and the program manager in Project Ocean communicated project status as one of their main tasks. Information about who do what, and current status of development tasks was formally exchanged in a daily meeting of the manager group (Project River) or weekly report to customer team (Project Ocean). In Project Tree, a gatekeeper subscribed for changes from the OSS community and informed management level about the status of the release.

In Project River, the team leader was the only person in Team Thames was informed about Team Danube’s work status from the customer. This information was not frequently circulated in Team Thames, but when a problem arises, such as a bug or possible delay of a joint task, Team Thames members were informed by the team leader: “I need to be sure that everyone got what they needs, nothing more. We often have indirect contact with [Team Danube] via customers. We almost always send emails with the copy to relevant product owners and vice versa. For some tasks, I am informed about [Team Danube] from the customer” (a team leader from Project River).

### A. Commonality across cases

To response to RQ1, we observed three common characteristics of boundary spanners in all cases. Comparison of boundary spanner roles and boundary spanning activities across four cases, we observed common characteristics and
activities of boundary spanners, as shown in Table 5. As a boundary spanner-in-practice has to develop both an inclination and an ability to be a legitimate peripheral participant in spanned groups [17], the team recognition and multiple area expertise is a result of these peripheral activities. We also agreed with previous studies about the decision-making and leadership as part of core capabilities of a boundary spanner [52]. Cataldo et al. observed that the formalized boundary spanners tended to contribute less to the development effort than the rest of the developer, while informal boundary spanners emerged from a highly productive developer [53]. Complement for this finding, we showed that informal boundary spanners contribution might not be properly recognized by the management level.

To response to RQ2, we identified task negotiation; conflict resolution, task information navigation and boundary object set-up as common boundary spanning activities of an in-practice GSD spanner. Interestingly, the ability of hiding information is found in an OSS project, which shows that global boundaries in OSS projects might not be transparent as it is reported in previous literatures. Moreover, all boundary spanners participate in setting up boundary objects, such as establishment of code convention and JIRA tracking tickets, yet the activity to enforce practices along with these boundary objects occurs only in one case.

B. Differences across cases.

Comparing four cases shows how characteristics and practices of boundary spanners vary across different global setting context. We discussed compounding effects of organizational roles on boundary spanners, differences between commercial and open source projects, small-size and large-size projects, inter-firm and inner-firm projects.

Compounding factors of organizational roles on boundary spanner characteristics: characteristics of boundary spanners are significantly impacted by organizational roles that they have. One would expect that if an individual in a boundary spanning role is a manager or a team leader, his/her “influence” would be drastically different from a person that performs the boundary spanning role but has non-decision-making organizational role. In Project River and Mountain, boundary spanners had an ability to influence team coordination across boundary, hence, we observed team configuration and management decision as a compliment coordination mechanism. In Project Ocean and Tree, the boundary spanners are a bridge engineer and a lead developer, which is limited in the ability to influence the team. However, we still observed an ability of making decisions on task breakdown, arrangement, and communication in these cases. Alternatively, boundary spanners in managerial roles can perform mechanic coordination activities, while boundary spanners in non-decision making roles are limited to organic coordination approach. Regardless of organizational role, boundary spanners are recognized by internal team members and external stakeholders. Perhaps, adoption of both formal communication via managerial roles and information communication with developers across boundaries leads to the wide-spread of this role [33].

Table 5: Commonalities and differences among cases

<table>
<thead>
<tr>
<th>Context</th>
<th>Spanner capacity</th>
<th>Boundary spanning activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commonalities</td>
<td>Internal team’s recognition</td>
<td>Task negotiation, conflict resolution, Task-related information navigation</td>
</tr>
<tr>
<td></td>
<td>Inter-features technical expertise</td>
<td>Task assignment</td>
</tr>
<tr>
<td></td>
<td>Task assignment</td>
<td>Boundary object setup</td>
</tr>
<tr>
<td>Management roles vs. non-management role</td>
<td>Decision making ability</td>
<td>Mediating status information</td>
</tr>
<tr>
<td>Commercial vs. OSS project</td>
<td>Work time and place flexibility</td>
<td>Comprehending and transferring tasks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Site visit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enforcing practice around boundary object</td>
</tr>
<tr>
<td>Small size vs. large size project</td>
<td>External team recognition, Knowledge of business logic and technical expertise</td>
<td>Task integration</td>
</tr>
<tr>
<td>Inter-firm vs. Inner-firm</td>
<td>Ability of team configuration</td>
<td>Facilitating exchanged practices</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sharing knowledge and expertise</td>
</tr>
</tbody>
</table>

Compounding factors of organizational roles on boundary spanner practices: one can also expect that coordination practices adopted by managerial boundary spanners are different from a non-management boundary spanner. However, we did not notice a systematic difference in coordination practices between these two roles. Except for Project Ocean, program manager devoted significant time for coordination activities, in other cases, coordination activities are carried on with other development activities.

Commercial and open source projects: OSS participants are not bound to company regulations and project deadlines as in commercial firms. In this case, a boundary spanner does not need travel and shift work time. A boundary object in Project Tree did not perform comprehending and transferring tasks. Therefore, OSS boundary spanner regulates development practices on boundary objects, which is not seen in commercial boundary spanners.

Small-firm project and large-firm projects: we noticed some differences between Project Mountain and the others. A boundary spanner in large-firm projects has less external team recognition and cross-discipline expertise. This might be cause of high level of specialization and well-established development processes in large companies. Integrating and communicating deliverables to management levels are also activities that are highlighted in large-firm projects.

Inter-firm and inner-firm projects: we also observed some differences between small and large organizational distance. Ability to configure a team is observed in an intra-organizational project as team configuration does not have to cross a large organizational gap. Sharing knowledge and expertise is harder in inter-firm projects. Interestingly, Project River shows that boundary spanners facilitate exchange of practices between competing partners. Perhaps, when a practice is perceived as a competitive advantage of the rival,
the practice would be adopted. This function of boundary spanners in inter-firm context is not visible in inner-firm projects [44].

VI. CONCLUSIONS

A lot of research work have focused on spanning activities across cultural boundary in IS and SE literature. In this study, we made contributions to current body of knowledge on team coordination in GSD by (1) characterizing characteristics of boundary spanners as a project coordinator, (2) identifying the boundary spanning activities that address interdependencies across organizational boundary.

Using the checklist proposed by Runneson et al [47], we evaluated our case study design. Many conscious decisions were taken to strengthen the validity of results. In this study, the interview guide was generated from theory and previous literature, which reduce the threat of construct validity. Besides, we performed the study in four different types of organizational context settings with variety of project size, global arrangement and product domains. Using both typical and extreme case reduces the threat of generalization. Credibility concerns about the confidence of the result. We adopted several threat mitigation strategies, such as sending interview transcripts to each interviewee for correction; involving three researchers on study design and review, and detailed documented steps of data collection, processing and analysis. Thus we believe that the resulting protocol will serve as a good means to replicate this study. Last but not least, we used multiple data sources and viewpoints to strengthen the evidence generated in our cases.

Our study has several implications for practitioners, i.e. project managers and team coordinators who would like to improve team-level coordination effectiveness. Firstly, the least requirement for a boundary spanner is to obtain internal team recognition, multiple technical expertise and ability of task assignment to be an effective coordinator. The most common activities done by these roles are negotiating and navigating task information, resolving conflicts and setting boundary objects. Secondly, it is critical for managers to be aware of the compound effect of boundary spanner and other coordination mechanisms, such as boundary objects. In a context where one of them is inhibited or ineffectively adopted, focusing on other mechanisms can help to compensate. Thirdly, our cases revealed that availability of boundary objects is not sufficient to make them an effective coordination mechanism. A governance policy across organizational boundary should be adopted to establish the consistent adoption of them. Fourthly, while boundary spanners are important for team coordination, they also face some issues, such as knowledge silos, role conflicts and stress. Therefore, the strengths and weaknesses of adopting boundary-spanning activities should be taken into account. Boundary spanner’s goals should be recognized and aligned with organization’s goal.

Future work can extend this study by (1) conducting a longitudinal study on addressing questions about “when boundary spanner leverage boundaries” or “when boundary object is effectively used?”, (2) elaborating our conceptual model and derive measurable hypotheses and (3) identifying how and what factors drive the decision-making procedure of boundary spanner when mediating task and information flow.

APPENDIX A

1. Please briefly describe your role in the project.
2. How do you receive the task from customer and other teams?
3. How do you communicate with customer and other teams?
5. What has been done to solve this problem? (Site visit? Collaborative tool? Sharing common process? Established standard in requirement/ code/ commits?)
6a. What is the role of a bridge engineer/ manager/ coordinator in collaborating task among teams? What kind of influence that they have in communication?
6b. What has done to coordinate teams? What are practices that help you to coordinate team activities?
7. What can be done to improve team coordination?

REFERENCES


23. E. Hochmiller, “The requirements engineer as a liaison officer in agile software development”, Workshop on Agile Requirements Engineering, Lancaster, United Kingdom, 2011.


46. Gopal et al. Coordination and Performance in Global Software Service Delivery: The Vendor’s Perspective.

47. P. Runeson, M. Host, A. Rainer, B. Regnell, Case Study Research in Software Engineering.


