Does Distance Still Matter? Revisiting the CSCW Fundamentals on Distributed Collaboration

PERNILLE BJØRN, IT University of Copenhagen, University of California, Irvine
MORTEN ESBENSEN, RASMUS ESKILD JENSEN, and STINA MATTHIESEN,
IT University of Copenhagen

Does distance still matter? Reporting on a comparative analysis of four ethnographic studies of global software development, this article analyzes the fundamental aspects of distance as depicted in the famous paper "Distance Matters." The results suggest that, although while common ground, collaboration readiness, and organizational management are still important aspects for distributed collaboration, the arguments concerning coupling of work and collaboration technology readiness need to be refined. We argue that in working remotely, closely coupled work tasks encourage remote workers to spend the extra effort required in articulation of work to make the collaboration function. Also we find that people in distributed software development have already made collaborative technologies part of their work, and individuals are comfortable with them; thus, collaboration technology readiness takes a different shape in this setting.

Categories and Subject Descriptors: K.4.3 [**Computer-Supported Collaborative Work**]

General Terms: CSCW, Distance, Global Software Development, Ethnography, Comparative Studies

Additional Key Words and Phrases: Common ground, collaboration readiness, collaboration technology readiness, coupling of work, closely coupled work

ACM Reference Format:
Pernille Bjørn, Morten Esbensen, Rasmus Eskild Jensen, and Stina Matthiesen. 2014. Does distance still matter? Revisiting the CSCW fundamentals on distributed collaboration. ACM Trans. Comput.-Hum. Interact. 21, 5, Article 27 (November 2014), 26 pages.
DOI: http://dx.doi.org/10.1145/2670534

1. INTRODUCTION

Since 2000, when Judith and Gary Olson published their famous paper with the catch phrase "distance matters" [Olson and Olson 2000], numerous research papers on distributed collaboration start by citing that working across distance is difficult. Although there is no doubt that the paper was first in bringing together research on collaboration across distance, much has happened since 2000 in terms of collaboration technologies, as well as in terms of people's experiences in working remotely and enacting collaboration technologies in their work. Thus, the initial coherent framework and later extensions of it [Olson et al. 2008; Olson and Olson 2014] have been shown to be useful analytical tools in the research on distributed collaboration. However, the development of new technical inventions introduced constantly into various collaboration setups continuously transforms the conditions for collaboration work practices, which

Author's address: P. Bjørn, Technologies in Practice research group, IT University of Copenhagen, Rued Langgaards Vej 7, 2300 Copenhagen, Denmark; email: pbra@itu.dk.

DOI: http://dx.doi.org/10.1145/2670534

means that it is essential that we, as Computer-Supported Cooperative Work (CSCW) researchers, regularly revisit the CSCW fundamentals to determine whether previous concepts and knowledge need to be updated [Schmidt 2009]. Therefore, the aim in this article is to ask critically: Does distance still matter?

At the time of the Olsons' paper, distance was conceptualized into four main concepts, which together explain how distance matters. These concepts are common ground, coupling of work, collaboration readiness, and collaboration technology readiness. Later, in 2008, a fifth concept was added to the distance framework: namely, organizational management [Olson et al. 2008]. We believe that it is time for us to reanalyze the basic factors presented when people work at a distance, especially since working remotely is no longer a special case, but is instead the new normal for workers such as software developers [Herbsleb 2007]. We need to periodically critically examine whether our basic CSCW understanding of what makes collaboration at a distance difficult continues to be the same or whether there are new factors or new nuances to the preexisting factors.

Being engaged in a large research project on global software development since 2010, the authors of this article have conducted four ethnographic studies of the work practices between IT developers involved in software development across different locations such as the Philippines, Denmark, India, the United Kingdom, the United States, and Germany. Although each ethnographic study is unique, they all relate to the concepts of collaboration across distance. Thus, we were able to analyze the concept of distance by challenging the framework through our unique empirical cases, as well as by comparing across the cases. Comparing empirical data across large ethnographic studies continues to be a challenge in CSCW research [Crabtree et al. 2013], and few have published such work [Balka et al. 2008; Boulus and Bjørn 2008; Avram et al. 2009]. In this article, we take up this challenge and present the results of four empirical cases of remote global software development.

This article takes seriously the call to question and analyze CSCW fundamentals [Schmidt 2002] when new collaboration situations emerge due to the introduction of new technological opportunities. We provide an exemplary method for how to conduct comparative analysis across ethnographic studies that can help us to further develop the field of CSCW [Schmidt and Bannon 2013]. Thus, the main contribution in this article is our results, which show that although common ground, collaboration readiness, and organizational management continue to be critical factors for collaboration across distance, the two factors of coupling of work and collaboration technology readiness need to be reconceptualized differently than they were 15 years ago. We find that closely coupled tasks make collaborators interdependent in their work by producing strong dependencies across the remote participants. These dependencies force dispersed participants to commit to engaging in the extra effort of articulation work, such as the frequent interaction and communication that is required when working remotely. Different loosely coupled work arrangements tend to push remote collaborators even further apart since there are no dependencies forcing them to interact. Finally, we found that although collaboration technology readiness was critical 15 years ago [Grudin 1994], today the situation is different. Today, collaboration technologies of all kinds are in common use, at least in the global software development domain, and therefore participants do not experience the challenge of collaboration technology readiness. We did find that collaborative partners have to spend time and effort in creating sociotechnical connections, including making collaborative technologies function across sites; however, this kind of collaboration technology readiness is more in line with relation work [Bjørn and Christensen 2011; Christensen et al. 2014]. Finally, we also include the recent extension of the distance framework (namely, organizational management [Olson and Olson 2014]) in our investigation of cases. Here, we point to how organizational management, including reflective processes of rotations within teams working remotely, is critical to the strength of long-term collaborative setups between organizations.

The article is structured as follows. First, we revisit the literature on distance starting from the "Distance Matters" paper back in 2000, and we fold in recent literature on distributed work. Through the literature, we develop the distance framework, which we then apply as an analytical lens in our comparative analysis. In the method section, we present the empirical cases and explain how we applied our analytical lens to the data we collected. The results section is structured thematically: we dedicated a section to each of the concepts: common ground, collaboration readiness, collaboration technology readiness, coupling of work, and organizational management. We end this section by summarizing the empirical observations. In the discussion section, we bring forward the literature by discussing and challenging each aspect of the distance framework through our empirical observations. Finally, we present our conclusions.

2. DISTANCE MATTERS

Understanding the basic nature of collaboration across distance and the role of collaboration technologies in such settings has been a keen interest for CSCW research for many years [Finholt and Sproull 1990; Mark et al. 2003; Gutwin et al. 2004; Nardi 2005]. One of the most canonical papers is "Distance Matters" [Olson and Olson 2000]. The majority of academic papers investigating geographically distributed collaboration begin their introductions by referring to this paper. Although there is no doubt that distance matters for collaboration, in this article we want to investigate whether distance still matters in the same way—or whether the last nearly two decades of technological interventions and social practices have changed the fundamental nature of *how* distance matters. However, before we go into the empirical findings from our cases, we need to take a step back and see how distance was first conceptualized back in 2000. There are four basic aspects of distance: Common ground, collaboration readiness, collaboration technology readiness, and coupling of work [Olson and Olson 2000]. We refer to these aspects as the *distance framework*. However, recently, the Olsons have added one more aspect to the framework—organizational management; thus, we include this concept as well. Here, we revisit each aspect individually.

2.1. Common Ground

Common ground refers to the knowledge that people share and that they know the other shares, a concept based on the grounding practice within communication [Clark and Brennan 1991]. Grounding is accomplished when collaborators engage in communication and express confirmation of comprehension through words or bodily movements. Grounding is made difficult when the risk of misunderstanding is high, a risk that increases by adding multiple languages and cultural practices, as well as dislocation of participants [Olson and Olson 2000]. Common ground has also been referred to as the *mutual knowledge problem* in distributed collaboration [Cramton 2001]. Several types of problems arise from the mutual knowledge problem including failure to communicate and retain contextual information, unevenly distributed information, difficulties in communicating and understanding the salience of information, differences in speed of access to information, and difficulties in interpreting the meaning of silence [Cramton 2001]. Communication across distance in the process of creating common ground and avoiding the mutual knowledge problem has been studied and investigated extensively, for example in terms of negotiation and commitment [Bjørn and Hertzum 2006], conflict [Hinds and Mortensen 2005], and communication breakdowns [Bjørn and Ngwenyama 2009]. In the most resent research, the common ground problem has been explained in terms of divergence and convergence in concepts and meaning

[Jensen and Bjørn 2012]. In this article, we critically examine whether common ground as a feature of collaborative practices continues to be a core problem for geographically distributed participants and thus still a design challenge for CSCW technologies and social practices.

2.2. Collaboration Readiness

Collaboration readiness concerns those aspects by which participants within distributed teams are ready to engage in collaboration activities together. Collaboration readiness has been related to incentive structures that reward participants based on their engagement with others [Olson and Olson 2000] or organizational structures that are instead supportive of competitiveness. Whereas collaboration readiness was originally created as a concept to capture the motivational aspects of collaboration, it is critical to note that other aspects of collaboration readiness are not closely related to motivation, but are instead related to the aspect of whether participants are ready to engage with others on a more general level. To explain this further, heterogeneity and diversity, which often exist within distributed collaborative setups, are often experienced as problematic in terms of collaboration readiness. To provide an example, if experienced software developers are placed in collaborative setups where their counterparts are junior and inexperienced developers, this might seriously impact the motivational setup for the senior developers and thus their collaborative readiness. The larger the "reach" within the organizational setup, the more problematic [Gerson 2008] it becomes. Interestingly, it has been suggested that it is not the degree of diversity that threatens collaboration, but instead the risk of creating subgroups within teams, when key attributes of members (e.g., language, professions, age, etc.) are correlated with geographical location rather than cutting across sites; this is also referred to as the *alignment of faultlines* [Cramton and Hinds 2005]. The presence of faultlines increases the risk of subgroup formations based on the geographical location of people, which is counter to creating collaboration readiness and also increases the risk of conflict. Other aspects, such as trust, commitment, and transparency, have also been found important for collaboration readiness [Mark 2002; Søderberg et al. 2013] and different patterns in terms of how trust is created [Jarvenpaa and Leidner 1999] or why people might act in terms of deceptions and persuasion in geographically distributed settings [Bradner and Mark 2002]. *In this article, we examine whether collaboration readiness is still an important challenge within geographically distributed work and, if so, in what ways.*

2.3. Collaboration Technology Readiness

Collaboration technology readiness refers to the difficulties faced in adapting, adopting, and bringing collaboration technologies into use [Olson and Olson 2000]. Technology readiness was identified back in the '80s as a complex challenge for collaboration technologies for several reasons. First, concrete experience with collaboration technologies was less common in organizations at the time, and important studies of the challenges in adapting technologies to organizations were pertinent [Orlikowski 1992; Orlikowski and Gash 1994; Grudin and Grinter 1995]. Adapting technologies required much work in renegotiating protocols for use [Bjørn 2003], as well as spending time to develop technological frames [Bjørn et al. 2006] and aligning technology [Majchrzak et al. 2000; Bjørn and Ngwenyama 2010]. Collaboration technologies clearly provide different types of challenges than do single-user systems. As Jonathan Grudin listed in his famous paper *Eight Challenges for Developers* [Grudin 1994], the challenges for collaboration technologies include the "disparity in work and benefit," "critical mass," "disruption of social processes," "exception handling," "difficulty in evaluation," "unobtrusive accessibility," "failure of intuition," and "the adoption process." Each of these challenges forms a particular set of conditions for collaboration technologies, which are different from single-user systems. Much has happened in the design and use of collaboration technologies, and, today, using collaboration technologies is routine, everyday practice for many. Therefore, in this article, we investigate whether collaboration technology readiness is still pertinent for collaboration across distance or whether we need to re-evaluate this factor when we think about design of CSCW systems.

2.4. Coupling of Work

Coupling of work refers to the characteristics of the collaboration work itself [Olson and Olson 2000]. Schmidt and Bannon write that at the core of the conception of collaboration is the *notion of interdependence* in work [Schmidt and Bannon 1992]. They explain how interdependence does not arise simply from sharing the same resources, but instead that "being mutually dependent in work means that A relies positively on the quality and timeliness of B's work and vice versa and should primarily be conceived of as a positive, though by no means necessarily harmonious interdependence" [Schmidt and Bannon 1992, p. 13]. People are involved in collaboration work when they, in their individual activities, change the state of the common field of work and thus have to act accordantly to the work of others. Due to the interdependencies within collaboration work, extra work is required by the participants to mediate, coordinate, and align their individual yet cooperative activities—namely, articulation work [Strauss 1988]. Although articulation work originally was studied as part of co-located work practices as in some medical practice [Strauss et al. 1985; Møller and Bjørn 2011] and control rooms [Harper et al. 1989; Heath and Luff 1992], articulation work is key to current research on distributed collaboration such as occurs in global software development [Boden et al. 2014; Matthiesen et al. 2014]. If we look back at the Olsons' paper, we find that the argument around coupling of work is that tightly coupled work is difficult to accomplish remotely and that people in the Olsons' empirical cases tended to reorganize the work in such a way that tightly coupled work was assigned to people who were co-located, whereas loosely coupled work was the preferred way of working remotely [Olson and Olson 2000]. This trend is also what we find in studies of radical co-location [Teasley et al. 2000], extreme co-location [Mark 2001], or in strategies to minimize interaction [Hertzum and Pries-Heye 2011]. However, interestingly, in the current research on distributed collaboration found in, for example, engineering [Bjørn and Christensen 2011] or software development [Christensen and Bjørn 2014], working remotely in closely coupled situations is becoming a routine practice [Herbsleb 2007; Esbensen and Bjørn 2014] wherein the information infrastructures embedded within the software task stipulate closely coupled work [Matthiesen and Bjørn 2015]. Thus, we need to analyze the fundamentals of coupling of work; therefore, *in this article, we examine the ways in which coupling of work impacts the collaboration practices across distance, questioning whether loosely coupled work across distance is the best organizational form.*

2.5. Organizational Management

Organizational management refers to the practices by which management activities are part of shaping the fundamental premises for collaboration without proximity [Olson and Olson 2014]. It includes aspects such as the structural setup of the organization, decision-making processes, legal issues, and managerial issues of supporting the collaboration across distance [Olson and Olson 2014]. The organizational context of distributed collaboration is a conglomeration of pieces, including organizational policies, norms, professional language, and procedures, that participants bring together from local organizational contexts and apply toward the distributed collaborative context [Bjørn and Ngwenyama 2009]. Thus, when people work across locations, and especially across organizations, the local organizational context (including local management practices) shapes the conditions for the collaboration across distance. For example, in a case reported in literature [Bjørn and Ngwenyama 2009] in which a team comprising participants from different countries meet regularly and travel across the global organization to collaborate, each local organizational context has its own travel policies, even though they are all part of the same large global organization. Thus, it was not possible for the team to plan combined travel with the aim of working during flights because booking flights is a local organizational process and cannot be organized globally [Bjørn and Ngwenyama 2009]. The importance of creating shared or at least partly shared managerial practices supporting collaboration across locations was further established in the impressive and much cited work on the Boeing-Rocketdyne case study [Malhotra et al. 2001]. In that paper, the authors point to management practices, such as creating a strategic setting and restructuring the work to accommodate the distributed setup without changing the core creative needs, as essential for success [Malhotra et al. 2001]. Managerial practices include technology alignment activities [Bjørn and Ngwenyama 2010], such as supporting the iterative reconfiguration of collaborative technologies to make these support the continuous emergent changes in work practices that tend to happen over time due the malleable features of distributed collaboration [Majchrzak et al. 2000]. Although the managerial aspects were not part of the original distance framework from 2000, our aim in this article is not to establish whether organizational and managerial aspects are still important—they clearly are; instead, our focus *is to establish how the managerial aspects were organized in our four cases so that we can include these reflections in understanding the critical factors to be included within the distance framework and what these entail.*

3. METHOD

To analyze the fundamental concepts of distance in collaboration to find out whether distance still matters and, if so, in what ways, we took advantage of empirical data that we collected as part of the large research project investigating Next Generation Tools and Process for Global Software Development (NexGSD). In NexGSD, we work with several industrial partners who are engaged in global software development across countries such as India, the Philippines, Germany, the United Kingdom, Denmark, and the United States (see also nexgsd.org). We present four empirical cases, each with different setups for global software development across distance: INIT, GLOBALSOFT, INDK, and SCANDIABANK. All the cases are based on ethnographic inquiry of the situated practices of collaboration within the organizations [Randall et al. 2007]. We wanted to understand how and why people engaged in software development and, although being located around the world, managed to work despite possible challenges. We wanted to know what these challenges might be and how they could inform the design of new collaboration technologies. Each case provides insights into the collaboration practices as they are organized in particular setups. Although each case is unique in itself, the cases also share similar aspects and patterns that we also investigated.

Conducting comparative analysis of ethnographic data material has been called for within CSCW research [Pollock and Williams 2010], with the aim of building on existing knowledge. However, only a few have conducted such cross-case comparisons [Herbsleb et al. 2005; Schmidt et al. 2007; Balka et al. 2008; Boulus and Bjørn 2008; Boden et al. 2009]. It is important to mention that the purpose of our cross-case analysis is *not* to create generalization in ethnographic studies [Crabtree et al. 2013]; clearly, ethnographic studies are valuable as single entities. Instead, we present an analytical method by which we are able to analyze particular theoretical concepts from literature and see how different empirical situations might respond. In this section, we present each of the empirical cases individually and explain the nature of the empirical data material we have gathered from each case. These data, in turn, have collectively served as grounds for our theoretical analysis of the conceptual bases for distance.

3.1. Empirical Cases

The four empirical cases were designed differently, each depending on the research question that we were interested in, the possibility of access to the field site, and the interest or willingness of the industrial partner. A summary of their characteristics is shown in Table I.

3.1.1. INIT Case. INIT is an Indian multinational information technology company specializing in software design, software services, business processes, and consulting. INIT operates out of 46 countries worldwide and serves large global clients in banking and insurance. In the period spanning December 2011 to February 2014, we were able to follow three large global software development projects within INIT. We visited INIT sites in India and the United Kingdom in December 2011, January 2012, January 2013, and February 2014. In total, we conducted 65 interviews with 42 people (at all levels of development from global delivery head and program manager to tech lead and programmer). Eight people were interviewed three times each, nine people were interviewed twice, and 25 people were interviewed once. The interviews lasted one hour, and all were transcribed. In addition, a few observations at the organization were conducted during the periods of interviews. However, we were not able to observe the work practices of developers. In total, three researchers took part in collecting the data.

3.1.2. GLOBALSOFT Case. GLOBALSOFT is a software development company that incorporated in Denmark in 1994 and handles all stages of specification, development, testing, and implementation. GLOBALSOFT is largely involved with developing public and governmental systems for Danish society, and it develops highly specialized solutions in close collaboration with representatives of the Danish government. The company employs a total of nearly 1,700 people: 1,400 are employed in Denmark and 300 in offshore locations including China, Switzerland, the Czech Republic, and the Philippines. The main empirical work we conducted at GLOBALSOFT was an in-depth investigation of one particular project between Danish and Filipino developers. Data collection was undertaken by a total of four researchers and took place from December 2010 to October 2013. We conducted more than 50 on-site observations in Denmark and the Philippines, surveying everyday practices and accumulating approximately 300 pages of reflexive notes. The data include 28 audio recorded and transcribed interviews (19 in Denmark, nine in the Philippines), each about an hour long. We spent a total of almost two months in the Philippines during four different visits (December 2010, July 2011, November 2011, and January 2012) and then four months in Denmark, where one researcher spent two to four days a week in the organization followed by a period of eight months when one to two researchers visited the organization one to two times a month. During data collection, we discussed initial findings with the workers in casual conversations, as well as in official presentations and workshops. Two researchers conducted four video recorded workshops in the Philippines and Denmark. Last, we had access to internal company documents and screen recordings.

3.1.3. SCANDIABANK Case. SCANDIABANK is a Danish financial enterprise that employs more than 20,000 people in Northern Europe. Since 2006, the IT division of SCANDIABANK has been collaborating with ITS—a well-reputed IT vendor working out of Bangalore, India—on handling, maintaining, and developing the IT services for the bank. From October 2012 to March 2013, two researchers followed the development work of a department within the IT division of SCANDIABANK and their Indian collaborators and counterpart team in ITS. We conducted fieldwork in both Denmark and India. In total, we conducted 12 interviews of varying length featuring people with different occupations and roles including system managers, task managers, IT developers, business developers, and software developer expats from the bank and located in India. In addition, we conducted 80 hours of observation of developers' daily work and practices. A daily field diary was kept, as well as reflective notes and video recorded memos that prompted reflective conversations and discussions in the field. Finally, we analyzed various internal department documents including outsourcing strategy documents, meeting agendas, organizational diagrams, and task descriptions. Two researchers took part in collecting the data.

3.1.4. INDK Case. INDK is a large Danish development company focusing on software solutions for the Scandinavian financial market. Parts of the company's mission have been outsourced to a major Indian software solution provider. For two weeks in August 2013, two researchers—one in Denmark and one in India—followed one distributed team in this setup. The team was a distributed scrum[1] team with members from both India and Denmark. We were able to observe the daily collaboration as it unfolded between the two sites both in terms of planned meetings and in the nonplanned interactions through Instant Messenger and email. In total, we conducted 12 interviews with developers, testers, and the scrum master on both sites. The interviews lasted 30 minutes to one hour. In addition, we conducted simultaneous daily observations of the work practices and all meetings between the teams.

3.1.5. Data Sources. It was a concern for us that, in three cases, data collection was done solely by Danish researchers traveling to the Philippines or India. The concern was related to the risk of these researchers being perceived as "controllers" or "representatives" from the Danish company collaborators and that interviewees would try to present "appropriate" narratives and leave out negative events and attributes. Although critical and loud voices were definitely more common when interviewing Danish developers compared to Filipino or Indian developers, we would say that, over time, we did hear several critical stories and experiences from Filipinos and Indians. This might have to do with the long-term relationships we created with each case, particularly GLOBALSOFT (3 years) and SCANDIABANK (5 months). The INDK case was different because the study was much shorter than the other studies (only 14 days) and thus the risk of being viewed as a "controller" was more pertinent. In the INIT case, three researchers conducted the interviews; one was of Indian origin whereas the two others were Danish. In this case, it was clear that we heard much more critical details in the third year of interviews compared to the first year, which was very related to the long-term relationships established with the interviewees. Table I summarizes the data sources from the four cases:

3.1.6. Data Analysis. We explored the five concepts of common ground, collaboration readiness, collaboration technology readiness, coupling of work, and organizational management as an analytical scheme for the data. We carefully went through the data, examining them for empirical observations that address the five concepts. We then pulled direct quotes and examples from the data organized by each case as single entities [Eisenhardt 1989], and these were then used to depict how the distance framework was related to each case. These pieces of theoretically enriched data snippets were

[1] Scrum is an iterative and incremental agile software development framework for managing software product development. It is based on the idea that success comes when structures enable teams to self-organize by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members.

Table I. Data Sources of the Ethnographic Fieldwork Conducted in Europe (EU), the Philippines (PH), and India (IN)

Empirical case	INIT	GLOBALSOFT	SCANDIABANK	INDK
Interviews	65	29	12	12
Observations	Few	50 sessions (each 2–10 hours)	80 hours	36 hours
Number of researchers	3	4	2	2
Length	3 years: Dec 2011–Feb 2014	3 years: Dec 2010–Dec 2013	5 months: Oct 2012–Mar 2013	14 days: Aug 2013
Workshops	None	4	None	None
Reflective notes	Yes	Yes	Yes	Yes
Document analysis	Yes	Yes	Yes	No
Sites visited	2 India, UK (Europe)	2 Philippines, DK (Europe)	2 India DK (Europe)	2 India DK (Europe)
In office[a]	No	Yes: PH/DK	Yes: India/DK	Yes: India/DK

[a] In office refer to the fact that researchers were assigned desk space in the company where they were able to sit and work, which also provides flexible interaction with the field during unplanned activities.

then used to challenge the distance framework to see if it was still applicable or whether we needed to rethink some of our basic knowledge about distributed collaboration.

4. RESULTS

4.1. Common Ground

Looking across the cases, it is clear that the practice of developing common ground was critical in all the projects. Sharing knowledge and knowing how the remote collaboration partners interpret and create meaning is critical to making the collaboration function well. In all the cases, developing common ground was found relevant in terms of the topic area or domain knowledge in the software projects, but also in terms of the processes and methods by which the collaborations were organized. It was evident that creating common ground was difficult to do in practice.

If we begin looking into the *difficulties in creating common ground concerning the topic area and domain* based on our cases, it is important to note the kind of projects that form our data. In all our cases, the work domain was software development for industrial organizations, such as developing IT systems to support particular practices within particular organizations or institutions. These practices could include food inspection, public insurance, banking, or digital governance, and the organizations or institutions could include national governance, global banks, or insurance institutions. Each domain of practice places particular requirements on the software developers to be able to understand the complexities and professional language of the domain. In some situations, domain-specific language entails some generic concepts. Developing IT systems for a global bank requires professional knowledge about banking in general. There will always also be local knowledge required, such as the national laws of banking or the organizational process within the particular bank that was the client (as was the case in INIT). Developing IT systems for pharmacies requires general professional knowledge about prescriptions, as well as particular knowledge about the role and work of pharmacies in the specific segment of society in which these IT systems are to be implemented (as was the case at GLOBALSOFT).

In all the cases we analyzed, developing professionally tailored language around the particular domain for the IT system was a challenge requiring particular efforts and activities to overcome. In the GLOBALSOFT case, for example, developers located in the Philippines had difficulties learning Danish food inspection practices, which required additional iterations of the system requirements and product descriptions. In the INIT case, the Indian company decided to hire some of the European governmental employees who had participated in writing the law of public insurance that formed the legal base for the IT system they were developing. The government of the European client country had decided to implement a mandatory insurance policy system that all companies had to sign up for. The idea was to overcome the increasing burden of the elderly population by ensuring that all citizens had pension savings. But it was new legislation and no one really knew how it would be implemented in practice. Therefore, the design of the IT system was very much a process of figuring out how to create, orchestrate, and design an IT system supporting a change in both society and law. Employing people who knew the legal bases for the system under development was the same as hiring those people with the most domain knowledge. By bringing these people on board, it was then possible for the Indian vendor company to execute events that created opportunities for grounding practices wherein the professional domain knowledge of the European government about the new legislation could be relayed to developers located in India. These events took the form of workshops and town hall meetings in which client representatives, INIT top management, and project members interacted with developers at the different sites of development. The project on the European public insurance project had around 300 participants involved in developing the system. The size of the team created an environment for constant interaction and offered a steady supply of knowledge about the project over the course of the three years that we followed it.

The importance of *common ground related to processes and methods* for developing was evident in all four cases, and the lack of common processes and knowing how to collaborate was problematic in both GLOBALSOFT and SCANDIABANK. Knowing how remote partners work, which processes they follow, and how participants on both sites interact and interlink their work was found important in all our cases. Also, it was evident that in all cases this level of knowledge was difficult to achieve for workers. In the SCANDIABANK case, the developers located in India clearly expressed the need to understand the processes followed by the developers located in Denmark in order to better align with them. However, the team in Denmark was not accustomed to following particular processes and actually found following formal processes constraining to their work. The Danish developers were used to knowing all about the IT system, as well as the business knowledge of the client bank, which meant that the undocumented legacy source code was perceived as a minor issue until now. Most of the Danish developers were senior developers who had built their careers at SCANDIABANK. They were accustomed to simply walking by each other's offices asking questions about the system if they needed, and they rarely relied on written documentation. This scenario changed when they began collaborating with developers in India. Processes and methods suddenly became very important. Working in a remote setting clearly requires common ground concerning the way of working.

Although this issue was never really solved at SCANDIABANK, the developers at GLOBALSOFT continuously engaged in activities and events supporting the development of common ground for processes. Different strategies were applied. For example, a new team tried different structures and models, but, finally, one of the system architects took it upon himself to develop and systematize a communication model to fit the work style of the developers in the Philippines. Scrum methodology served as a basis for developing common ground on processes in the INDK case, in combination with other routine practices of daily interaction across sites. It was evident that the processes were based on quick turnaround, where remote participants reviewed each other's requirement specifications and also brought in the customer to ensure common ground.

Clearly, common ground was important in all four empirical cases and, as such, played an important part in understanding collaboration across distances. However, it was also clear that there were two types of common ground that were required in all four cases. The first was related to the topic domain of the IT system and the second to the work processes and practices across sites. We might consider dividing common ground into more precise categories that could encapsulate the complexities and diversity of common ground important for collaboration across distance.

4.2. Collaboration Readiness

Collaboration readiness is concerned with how motivated remote partners are to engage with one another. It was an important aspect in all our four cases. Interestingly, by examining our cases, it became clear that when we look at software development, collaboration readiness could take different forms depending on the organizational structure of the collaboration:

- Is it outsourcing or offshoring?
- Is it complementary businesses collaborating?
- Or, is it competing business organizations within the same profession?

Also, we found that, in most cases, participants at one site were more ready than participants at the other; thus, understanding collaboration readiness is not simply rating along one continuum, but instead understanding complex relations across several continua.

We begin by looking at the case in which the basic setup comprised outsourcing. It was very clear at SCANDIABANK that the Indian employees working for the outsourcing company were hired within a global organizational setting where collaboration across distance was the expected way of working. This meant that the developers located in India were well aware of their Danish counterparts: They knew their names, what time it was in Denmark at all times, and generally directed their attention toward their remote colleagues. In contrast, in Denmark, the developers had been hired into the IT department of the bank and were accustomed to directing their attention toward the bank, the banking business, and the local context of the bank in general. When SCANDIABANK decided to change this setup and engage with outsourcing, this clearly challenged the Danish developers' perception of their work, forcing them out of their comfort zone of speaking Danish and also running counter to prescribed methods by redirecting their attention to their new remote colleagues. Thus, whereas the developers in India could be said to be ready to engage with their Danish colleagues, the developers in Denmark were not all ready. Because of this, we cannot simply state whether collaboration readiness existed in the SCANDIABANK case because it depends on which part of the collaboration setup we examine. Clearly, the developers in India were much more ready than the Danish developers because the Indian developers were hired into a company that was born global.

If we are to examine the outsourcing setup in more detail in our cases, we find that the initiation of global software development in three of our cases (SCANDIABANK, INDK, and GLOBALSOFT) was originally based on the idea of moving work from one location to another, mainly to save costs. Thus, developers located in Denmark initially had legitimate reasons to be concerned about losing their jobs to developers in India or the Philippines. Such competitive situations have the potential to make the Danish developers less interested in collaborating and thus less ready to collaborate. However, in the three counterpart organizations in our cases (two in India and one in the Philippines), this was not the case. For them, working with remote colleagues was an expected part of the business because their jobs were born global.

However, there still remained clear differences among these three cases that affected the collaboration readiness of the Danish developers. In 2006, GLOBALSOFT initiated an outsourcing venture by collaborating with an IT company located in Manila. This change was initiated because the CEO of the Manila company was of Danish origin and educated in Denmark, but had been settled permanently in the Philippines for many years. Thus, the beginning of the venture into global software development for GLOBALSOFT was thought possible due to the mediation of the CEO. In 2009, after three years of collaboration, GLOBALSOFT bought the Manila company, changing the organizational setup from outsourcing to offshoring. As part of this change, all employees of the Philippine office were kept on, and the previous CEO became part of the global management team at GLOBALSOFT while still located in Manila. This organizational change also meant that the contractual agreement within the particular projects was changed. First, prior to the switch to offshoring, the developers often found themselves taking attention away from solving tasks to spend time discussing the number of hours required for specific tasks. After the switch, there were no longer discussions concerning the internal divisions of hours spent on tasks. This meant that the collaboration readiness at GLOBALSOFT as a whole was improved by the organizational setup. Although the Danish developers still expressed difficulties during our final interviews in the fall of 2013, the change to offshoring was seen, overall, as a significant improvement to the collaboration.

Second, although the main business in both the INDK case and the SCANDIABANK case is electronic payment and banking and *not* software development, banking and payment practices are extremely dependent on IT systems and thus cannot easily be separated. To this point, SCANDIABANK is the largest company to hire IT developers in Denmark, despite being a bank and not an IT company. However, the main difference between the two companies we analyzed is that the IT developers at SCANDIABANK work for the bank whereas the IT developers at INDK develop IT systems for external clients. This discrepancy between in-house development and client development also created differences in the developers' level of job security compared to their remote colleagues. Still, the risk of job loss existed in both cases and created more difficulties in terms of collaboration readiness. In the INDK case, half of the collaboration partners located at the Danish site were exchanged from the Indian site, and thus the partners in the global scrum meetings had a significant number of Indian developers at both sites (Denmark had four participants total: two Danes and two Indians; Mumbai had five participants total: all Indian). The point here is that, out of a team of nine participants, only two were Danes whereas seven were from the outsourcing company. This power in numbers has the potential to tip the balance in the relationship in terms of knowledge about the technical system being created. These four different organizational setups (outsourcing or offshoring; in-house or external development) provided different conditions for collaboration readiness in the three cases, and it is clearly important to distinguish among their differences and reflect on the organizational setup when determining collaboration readiness in remote work.

The fourth case, INIT, is different in that the collaboration with the European client is not based on offshoring or outsourcing per se. Rather, INIT is a global IT vendor of Indian origin that negotiates and creates contracts to develop IT systems directly with European, Asian, or American clients. As such, we have a case of external IT development conducted by a global company. Most of the IT development is done in India. However, over the course of our observation, the teams were never co-located in India but, in most cases, were spread across several sites within India. These sites included Bangalore, Mumbai, Chennai, and Calcutta—places that are fairly different from each other in terms of location, local languages, and culture. It is important to note that India has 30 different local languages, each with more than 1 million native speakers, and that it is a pluralistic, multicultural, and multiethnic society. In addition to having sites in several cities in India, INIT also has several locations within each city. In addition, each project will always have at least one or more locations within Europe, Asia, or America.

The dispersion of the company's developers aims at expanding the availability of resources, competencies, and levels of quality and also allows the company to be prepared to continue work in the event of attacks or bombings, war, or natural or other disasters. In many cases, the work that INIT does for their clients provides critical systems that can never be shut down, making duplication and redundancy across sites an important strategy. What is striking in the INIT case compared to the three others is that geographical distance is not considered problematic in any way and is seen as a resource rather than a constraint. When INIT picks its teams (within the three projects we looked at, the teams each numbered about 300 people), they know where they will have access to quality testers, technical experts, user interface designers, and more and thus base parts of the project in those locations. In some locations, INIT anticipates that the project will need to expand—say, for example, in two years time—and also takes this factor into account. For example, whereas the team in Chennai at the beginning of the project was not the largest in mobile applications, it had the capacity for expansion to a strong team of an additional 100 people within two years' time. In this way, the INIT case represents a global setup with a high degree of collaboration readiness across distance because this is the basic premise of the work. The motivation for working remotely is salient in all aspects and practices within the INIT case and in such an embedded way that the company's employees do not even notice it or remark on the issue.

When INIT sets up collaboration with its clients, it is very aware that it takes time and effort to make such collaboration function well. As part of its strategies to ensure its client is collaboration-ready, INIT frequently brings clients to the different offshore locations where the development is ongoing, as well as provides a high level of transparency in its work. For example, INIT shared possible problems during the initial launch of an IT system for insurance in a European country, calling up the client in the middle of the night to resolve issues. Complexities always arise in large IT development projects. By creating strong strategic partnerships based on commitment, transparency, and trust, the INIT case demonstrates how collaboration readiness is important, but requires much work and effort.

Looking at our cases, we clearly find evidence of the importance of collaboration readiness. However, the diversity in which collaboration readiness emerged in our data could indicate that the generic concept of collaboration readiness might be too broad to really embrace the complexity of distance and that we should consider, to some extent, the notion of collaboration readiness taking into account organizational setups, client bases, core business, and the size of the organization. Being collaboration-ready is no longer just related to motivational issues, but also includes other aspects of being ready to work remotely.

4.3. Collaboration Technology Readiness

Although the previous two aspects of distance (common ground and collaboration readiness) are still important when we examine our empirical cases of global software development, the evidence for collaboration technology readiness showed different results. When we first initiated our work in the NexGSD research project, we had the goal of studying the collaboration practices of global software development with the aim of designing new collaboration technologies to support the problems we uncovered. Thus, for each case, we examined their use and issues concerning technologies and artifacts to identify important new areas of design. However, participants clearly expressed in their interviews that there were no problems for which they wanted new technology as a possible solution. They all said they had the technology, they knew how to use it, and the last thing they needed was new technology. This might be particular to the domain of software development, where all participants were technically strong employees who paid attention to new technical inventions and knew about new possibilities. However, it also presents an interesting finding in terms of collaboration technology readiness. Today, collaboration technologies have been around for many years, are used often, and have thus become conventional in use. This trend could reduce the usefulness of assessing technology readiness for distributed collaboration. This could, of course, also be an issue that is characteristic of the kind of cases we studied because IT developers are considered high-tech experts with competencies in applying such technologies. Our results could have been different if the professional domains of the cases had been less technical.

The collaboration technologies[2] used (to varying extents) in all the cases were:

- Source-code environments
- Shared document spaces
- Social media
- Email
- Calendars
- Video conferencing
- Instant messaging
- Screen sharing
- Telephones

Interestingly, in the INIT case, collaboration technology readiness was the highest with the telephone, which was also the most often used and preferred technology. Here, instant interaction using the telephone across geographical distance was seen as the most stable technology. Thus, simple phone meetings, with shared desktops across several dispersed participants and without video feed, was the preferred way of working. In the other three cases, video conferencing locations were used much more often. However, the instability of the technology was always a hindrance, and thus much effort and work went into making connections across sites. For example, during one of our observations in the GLOBALSOFT case, we followed a team of three people around for about an hour in their attempt to locate an available video conferencing room where the equipment worked. After several failed attempts spent entering video rooms without access to Internet, without an available remote control, or already booked by others, they had to postpone the meeting. These observations highlight that collaboration technology readiness is more an issue of organizational practices and unstable technology rather than the lack of knowledge about the technologies.

Although we could consider the extra work required in making the collaboration technology function a part of what determines collaboration readiness, this does not completely capture the kind of work that went into making these technologies connect with remote partners. It was not about knowing how to operate the technology, nor was it about knowing whether the collaboration technologies existed; instead, it was about the work of making collaboration technologies function in practice. Thus, aspects other than collaboration technology readiness are required to conceptualize the use of collaboration technologies.

[2] To mention some of the different types of collaborative applications in use in the four cases, the list includes Microsoft Test Manager, Microsoft Communicator, Microsoft Lync, Sharepoint, HP Quality Center, Visual Studio – Integrated Development Environment, Team Foundation Server, JDeveloper, Subversion, HP Application Lifecycle Management, SAP, and Lotus Notes/IBM Notes.

4.4. Coupling of Work

Coupling of work has to do with the task and the organization of the work. Interestingly, when we examine the data across all four empirical cases, it is evident that closely coupled work was important in making collaborations function at all. Closely coupled work *forced* remote participants to frequently interact because they were not able to do their work in any other way. In addition, the closely coupled work had a side effect of making remote participants know more about each other because of frequent interaction. Thus, participants experienced a stronger collaboration in situations where the task was closely coupled. In all four cases analyzed, participants clearly expressed that closely coupled work was preferred across distances.

In the INIT case, closely coupled work was executed both across distances between the company's various sites within India, as well as within the collaboration between the Indian IT vendor and their clients' sites around the world. This close coupling was organized through strong project governance, with weekly meetings among top managers to follow the project in detail, as well as at the lower levels through different types of daily and weekly formal meetings, and, importantly, through casual exchanges across time zones and distances using the phone. For example, one of the IT developers at a European location for one of the projects explained that he would be in contact with his counterparts in Bangalore several times a day by simply picking up the phone to resolve an issue. These informal, yet frequent, daily interactions were also found in the INDK case. One of the key participants from the India site located at the Danish site was responsible for choosing which team members to work with in Mumbai and had daily contact with Indian counterparts outside the daily scrum meeting. Frequent interaction was critical in both cases. The applied method was informal interaction and use of the phone. It should be noted that, in both these cases, the partner in Europe was an employee of the Indian site who was placed onshore for a period of time and, therefore, was quite familiar with counterparts located in Bangalore or Mumbai. Although this example clearly supports closely coupled work in remote settings, this was not the only place where we detected such an arrangement.

One of the project participants at INIT who was working with American clients and organizing teams across several Indian sites explained to us how she had managed to remotely lead her teams for the last three years only using the phone—never meeting her counterparts through video conferencing or in face-to-face exchanges. Still, she managed to work closely and organize the collaboration. She was in frequent (daily or weekly) contact with team members. We also detected the close coupling of work across the European clients and the Indian IT vendor in the INIT case. In that case, frequent interaction was required both on daily and weekly bases, in which the European client would interact with the Indian counterpart on all sorts of matters. When asked whether such close coupling was difficult, participants explained that it was important to constantly keep each other in the loop and was required for the project. Thus, the argument here is that it is the *nature of the task* that forces participants to engage in frequent interaction because, without such pressure, participants would not spend the effort required to interact.

In the GLOBALSOFT case, we also detected that closely coupled work was essential to force participants to interact frequently and thus make the collaboration function. In that particular case, the team, dispersed between sites in the Philippines and Denmark, explained how their collaboration became "cloudy" when they had less frequent interaction because the nature of task changed. Also, we saw how they had to reintroduce different types of daily and weekly meetings when they realized the tightly coupled nature of testing software. In the testing phase, they decided to have daily meetings (in the morning in Denmark and the afternoon in the Philippines) without agendas simply to keep each other in the loop and ensure that employees at the Danish site could continue the work of employees on the Philippine site when the workday in the Philippines ended but continued in Denmark. Likewise, in the INDK case, daily scrum meetings combined with daily informal phone meetings were common practices in making the collaboration function.

Taking a step back from these cases, one might argue that it is simply the type of work involved in global software development that requires such frequent interaction. However, although we agree that the type of task definitely impacts whether closely coupled work is required or not, our empirical data also point to other aspects that determine this. Close coupling made the remote partners pay attention to one another and, as a result, make efforts in getting to know their remote colleagues, thereby supporting the development of commitment, trust, and transparency. Close coupling also supports the articulation work required in all types of collaboration. Our data suggest that close coupling thus has additional benefits to supporting collaboration across distance because it forces people to pay constant attention to remote co-workers because information is required from them. In the SCANDIABANK case, Danish developers engaging daily with remote colleagues in Bangalore tended to increase the attention paid to and interest in their remote partners, thus further supporting the collaboration. In contrast, Danish developers who did not interact frequently with remote colleagues continued to be frustrated with the global setup, and they did not change their way of working. This suggests that closely coupled work tasks, in which a high degree of effort is required to handle the articulation work, pushes people to interact frequently because changes in individual activities (e.g., source code writing) cause changes in the common field of work (e.g., the common source code repository). Individuals cannot do their tasks without frequent interaction despite the lack of proximity.

Looking across the four cases, a clear pattern emerges when the work task requires closely coupled coordination. In those cases, the participants invest time and effort in their remote partners by engaging in frequent interaction because the task requires it. Thus, teams know more about each other, engage with each other, help each other, and develop strong engagement despite the geographical distance. By contrast, we found that in those situations where the task did not require frequent engagement, participants knew less about each other and felt more detached from the whole global collaboration setup. This suggests that closely coupled work in geographically distributed settings entails specific practices that support collaboration. Thus, this finding challenges the perspective that loosely coupled work is a better way of organizing distributed collaboration. Instead, our findings suggest that the close coupling of tasks in remote settings produces certain work conditions that beneficially support collaboration across distances and that these conditions do not exist in loosely coupled settings. Furthermore, our findings question whether strategies of deconstructing work tasks into smaller components to avoid coupling is the right approach for global software development.

4.5. Organizational Management

Examining organizational management in the four cases studied, one key factor is found across all: the ability for management to ensure that the global collaboration is not dependent only on a few key people but, rather, that the interaction across sites is divided across several people. In the GLOBALSOFT case, the manager of the Manila site in the Philippines was very aware that developers located in Denmark tended to ask to work with the same people repeatedly. This meant that the collaboration across sites was extremely dependent on a few so-called *power houses*, workers who where technically very capable and very experienced in working with the developers in Denmark. The developers in Denmark specifically asked for these power houses not only because they were competent (many developers at the Manila site were competent), but more so because they had come to know the Danes. This also meant that the manager in the Philippines found that a few of his employees were constantly overcommitted to projects and tasks, which created bottlenecks, while the other employees were idle. To accommodate this issue, the manager in the Philippines conducted a reorganization of the Manila site. This reorganization concerned clustering developers (both juniors and seniors) into teams, which meant that employees at the Danish site could not ask for specific people (power houses) but could instead ask for specific competencies or teams. In this way, the manager in the Philippines was able to create a rotation scheme that ensured that new people interacted directly with developers in Denmark, at the same time making it possible for his staff's overall experience to grow within the organization.

Rotation of staff was found critical in all the cases. Growing a global software development career often includes spending time either on-site or offshore for periods of time. In the INIT case, one of the managers who was responsible for around 300 people located at different sites within India explained that one of his sites where the team worked best was one where the employees had taken part in developing a rotation schedule for years to come. The schedule ensured that everyone with the desire to had the opportunity to spend six months on-site in countries such as England, Australia, China, the United States, or Germany. This practice helped retain people longer in the outsourcing organization. However, rotation across sites is not only important for outsourcing employees; it is also important for developers at the onsite locations or in the "mother" companies such as the INIT office in Denmark. In the SCANDIABANK case, it was clear that the few people who did travel and spend time in Bangalore made a shift in their attitude and perspective on working in a globally distributed setting, and they expressed that travel was an important learning experience. However, there was a restricted budget for Danish developers to work with those in India, and few employees had this opportunity to do so. In the GLOBALSOFT case, the organizational change from outsourcing to offshoring also meant that the travel policy changed, thus making it easier to travel from both sites. This increased the number of employees who traveled; however, some developers in Denmark actually preferred *not* to travel due to family obligations. Still, it was clear that travel was seen as a necessary aspect to making the collaboration work. This was a shift from the previous way of thinking when it was only management who traveled, and traveling was perceived as an extra bonus people received. In the INDK case, developers from India were in Denmark over longer periods of time, and it was clearly important for the work. The estimated preferred time period for developers to spend at the "other site" tended to be in the range of six months because this would provide enough time to get to know people and practices while still being brief enough in terms of maintaining a regular rotation schedule for a larger group. In situations where developers spent one or more years abroad, other developers were concerned that they would not get a turn to travel before it was too late for them to take advantage of it.

Looking across the cases, we found that routine rotations of boundary spanners across sites is critical for establishing a pool of high-quality people with both technical and organizational knowledge, as well as excellent international communication skills. To make collaboration across distance viable in the long term, people need to be able to grow in the organization. In the global software development field, this involves, in many cases, the ability to "go onsite" or "offshore" for a period of time, typically six months. Therefore, it is critical that such opportunities continuously exist.

4.6. Summary of Empirical Findings

We have provided the results of our analysis of the five aspects of distributed work—common ground, collaboration readiness, collaboration technology readiness, coupling of work, and organizational management—across our four empirical cases. We found that although common ground, collaboration readiness, and organizational management are important aspects of collaboration across distance, we could consider developing more detailed concepts to capture the complexity of real-life cases of global software development. Concerning collaboration technology readiness, our cases suggest that the current perspective is inadequate to fully capture the work required to functionally operate technology. In the past decade, our experiences with collaboration technologies have changed the nature of technological readiness: On average, individuals are more technically prepared and, particularly in the field of software development, are technically adept. Technology readiness in global software development is therefore not about being aware of new technical opportunities but is instead an issue of making stable technologies that can compete with the well-established reliability of the telephone. Finally, our cases demonstrate how closely coupled work supports collaboration across distance in important ways by forcing participants to interact frequently and thus ensuring that they exert extra efforts in articulation work despite being geographically distributed.

5. DISCUSSION

We set out to critically examine whether *common ground* as a feature of collaboration practice continues to be a core problem for geographically distributed participants and thus still comprises a design challenge for the development of CSCW technologies. Our empirical data clearly demonstrate how developing common ground is still a challenge and an important aspect of collaboration. In all our cases, developing common ground was seen as difficult, and much effort and energy was put into facilitating grounding practices in different ways. In this way, our empirical data support previous research on the importance of grounding in communication [Clark and Brennan 1991; Olson and Olson 2000]. However, our data also show that the difficulties in creating common ground in collaboration across distance could be divided into two different types: namely, (1) the issue of developing shared knowledge concerning the professional domain knowledge related to the IT system under construction and (2) the issue of developing common ground concerning the joined processes and practices of how to collaborate. Although the first issue concerning domain knowledge is a general problem for all kinds of software development [Herbsleb 2007] and, as such, is not particularly related to collaboration across distance, the second issue of identifying, creating, maintaining, and applying processes and methods that support global collaboration is another type of problem. Common ground around people's collaboration processes tends to be created through local interaction, wherein participants observe the work patterns of others and act accordingly. As examples, this was the case with both SCANDIABANK and GLOBALSOFT. However, without physical proximity, partners have to find ways other than observation to ground their practices in terms of how to collaborate. Examples of such ways to learn and develop common ground around work processes are the global scrum practice followed by INDK and the highly structured process orientation at INIT. At INIT, explicit organizational processes for structured documentation of source code or hierarchical decision making are examples of practices that serve to create common ground across the multiple sites (with several locations in India and across the globe), as well as across large-sized teams (e.g., of 100–300 people). In this regard, the design challenge for how to support the development of common ground in terms of work practices is still a vital area of research.

Previous attempts to address the issue by providing theoretical concepts such as "social translucence" [Erickson and Kellogg 2000], "workspace awareness" [Gutwin and Greenberg 2002], or "documentscape" [Christensen and Bjørn 2014], therefore, could still present viable methods to be addressed in the design of CSCW systems.

We wanted to examine whether *collaboration readiness* continued to be an important challenge within geographically distributed work and, if so, in what ways. Across all the cases, our data clearly demonstrate that collaboration readiness was an important challenge in making the collaboration successful. Across all the cases, collaboration readiness clearly serves as a relevant means for examining distributed collaboration. Investigating the ways in which collaboration readiness was important, we actually discovered new dimensions of the concept. First, it was clear in all our cases that a collaboration setting is not a single entity that can be more or less collaboration ready. Instead, each of our cases demonstrates how the distributed collaboration setup is a multiplicity comprising a diverse and complex entity of sites, organizations, competencies, expertise, and politics. Prior research has pointed to the virtual team context as a conglomeration of what each participant brings to the table from his or her diverse local contexts [Bjørn and Ngwenyama 2009]. Our study takes this thinking a step further and finds that collaboration readiness within globally distributed collaboration settings is grounded within the conglomeration of participants and thus is not a singular continuum but a multiplicity of relations. Some of the relations make up the organizational setup (e.g., the contractual agreement) and whether it is based on outsourcing or offshoring collaboration efforts.

It was clear in the GLOBALSOFT case that the change from outsourcing to offshoring was a major step in supporting collaboration readiness. Also, it was clear that collaboration readiness is not only the organizational setup, but also includes relations such as subgroup dynamics [Cramton and Hinds 2005] and whether the collaboration partners are able to span heterogeneity and diversity [Gerson 2008] that tend to follow geographical locations as faultlines. Bridging these faultlines entails enabling conditions wherein new subgroups can emerge across gender, nationality, geography, seniority, and organization. Distributed collaborations, wherein participants are able to create subgroups across these faultlines, are much more likely to be collaboration ready. However, creating diverse subgroups across geography, nationality, and organizations entails commitment, trust, and transparency [Søderberg et al. 2013]. It takes time and effort to develop such organizational situations. We saw this, for example, in the INIT case, where participants clearly spent a lot of effort in creating collaboration readiness in both offshore and onshore locations. But similar practices were also evident at GLOBALSOFT, where the collaboration readiness of the Danish counterparts changed over time. And, finally, at SCANDIABANK, creating collaboration readiness from the Danish location was very difficult. Collaboration readiness is still an important analytical dimension when investigating geographical distributed collaboration. However, our data suggest that the analytical dimension needs more nuances to conceptually capture the multiplicity of relations critical to understanding when examining global work.

In the early '90s, lack of *collaboration technologies readiness* created certain challenges for developers and designers [Grudin 1994], as well as for organizational implementation and adaption [Orlikowski 1992]. It is clear that one could not have "instant collaboration" by simply implementing technology. Instead, collaboration depended on people's technological frames [Orlikowski and Gash 1994; Bjørn et al. 2006] or windows of opportunity [Tyre and Orlikowski 1994]. Furthermore, after each implementation, organizations could expect a drop in productivity before the technology became accepted [Grudin and Grinter 1995]. Clearly, the analytical dimension of collaboration technology readiness [Olson and Olson 2000] made sense, due to the lack of experience people had at that time with such technologies.

Interestingly, this kind of collaboration technology readiness was not evident in any of our cases. All the participants knew about the collaboration technologies available, they all knew how to use them, they chose not to use some of them, and generally expressed how the only thing they really did not need was new technologies. This puzzling data made us wonder whether technology readiness was not an important dimension because the fundamental basis for global software development is precisely this technology. We found that the issue of technology use was still relevant but in a different way. In the past almost two decades, the rise in use of various collaboration technologies, both in the workplace but also in the private sphere, has created increased knowledge about such technologies and has reduced some of the problems of collaboration technology readiness. Using collaboration technologies has become a common, everyday practice in the profession of software development, and the issues that remain have more to do with the instability of technologies rather than with knowledge and expertise in applying them. In all the cases, it was evident that the issues concerning collaboration technologies concerned the increased add-on work of locating spaces for using the technology (e.g., the video room) where the technical devices were present (e.g., remote controls) and the basic infrastructure (e.g. Internet, cables, etc.) functioned properly. Such issues of collaboration technologies as an analytical dimension when studying distributed work are relevant and important. However, the simple concept of “readiness” does not really capture this because “readiness” is not just a mental construct, but is, instead, a practice of making the technology work. Thus, other concepts, such as relation work [Bjørn and Christensen 2011; Christensen et al. 2014], which entails the extra work required to create sociotechnical connections within collaborations and is fundamental to conducting articulation work, could be more suitable. It is important to mention that fundamental new concepts for technology design might still require a steep learning curve even though this was not the case in our empirical data; for example, recent work on cloud computing report how users have difficulties in comprehending the conceptual use of cloud computing technologies [Voida et al. 2013]. However, we did not see any such technology readiness in our empirical work; thus, our data suggest that the design challenges for collaboration technologies are stability, accessibility, and availability of collaboration technologies supporting global work rather than technology readiness.

In the past decade, there has been a tendency to recommend *loosely coupled work* arrangements rather than *closely coupled work* in distributed work. The thinking has been that the fewer interdependencies in the work task, the more easily it will be handled across distance [Olson and Olson 2000]. Strategies such as minimizing interaction [Hertzum and Pries-Heye 2011] or deconstructing tasks into smaller tasks [Grinter 2003] have been suggested as relevant strategies for global work. Interestingly, our data from all our cases point to the benefits of closely coupled work tasks as the organizing principle for remote work. Analyzing this result more closely, we began to question what closely coupled work arrangements accomplished for the remote collaborators, and the results pointed us back to the aspect of interdependencies.

Collaborative work is defined by the existence of dependencies in work [Schmidt and Bannon 1992]. People collaborate when participants, in their individual work, change the common field of work and are dependent on the work of others and thus act accordingly. In *loosely coupled work*, there are fewer dependencies, and, therefore, participants are not forced to frequently interact with each other. Especially in geographically distributed situations, workers will not have the chance to meet informally by the water cooler. As a result, there is the risk of co-workers losing connection with each other. For example, at GLOBALSOFT, we found that in those situations in which participants were not required to have daily or weekly meetings, they experienced the collaboration as becoming opaque and cloudy. Previous studies have pointed to difficulties in the interpretation of silence [Cramton 2001]. Our data could suggest that in loosely coupled work participants risk distancing themselves from each other rather than working closely together due to the prevalence of "silence" between them. When the work task on its own introduces closely coupled tasks, as in the case of scrum teams at INDK, this forces participants to interact and often pushes the project forward while encouraging familiarity among co-workers. In other situations, as at SCANDIABANK, where the work task did not force close coupling, the lack of frequent interaction was thought to be problematic by participants. In that case, those participants who did not have to interact frequently clearly perceived that their fellow colleagues who interacted more with remote colleagues often had a much better foundation for global work. Also, the preferred way of working in INIT was to interact several times a day by simply phoning the remote partners. Time zone differences impact the level of ease in simply staying in contact across sites. Frequent interactions made the workday more stressful by requiring participants to answer emails early in the morning and be available on the phone late in the evening. This is a concern; however, the issue of global work and occupational health is outside the scope of this article. Clearly, frequent interactions support collaboration [Herbsled et al. 2000; Herbsleb and Mockus 2003]; however, our point here is that frequent interactions do not happen automatically in the busy life of a global software developer. Rather, specific conditions need to exist that define which developers will spend time and effort on communication; our study highlights that such conditions are implicitly embedded within the nature of closely coupled work tasks and thus force people to interact.

The motivations for engaging with remote colleagues and collaborative technologies have, in previous studies, been linked to the issue of disparity in work and benefits [Grudin 1994]. Here, the point was that collaborative technologies often require additional work from individuals who do not perceive direct benefit, and this risks jeopardizing the use of the technology. Similarly, our argument on the conditions for engaging with remote colleagues stipulates that such engagement requires additional work from individuals—and if these individuals do not perceive direct benefit (in terms of direct relevance to solving their individual work tasks), they are less likely to spend the time required. Surely, good conditions for frequent interaction can be organized in many different ways, as we saw at GLOBALSOFT, where managers initiated agendaless meetings with the aim of forcing frequent interaction even in situations where the task in itself did not entail close coupling. However, our point here is that *the nature of closely coupled tasks* in global software development does entail frequent interaction through the asynchronous nature of working across time zones, thus creating a situation wherein participants get to know each other (as well as their remote colleagues' working habits). This creates the opportunity to strengthen geographically distributed collaboration.

It is important to emphasize that we do not argue that increased coupling is a universal general solution for all kinds of distributed work. Instead, we provide empirical evidence that clearly questions the "generally" stated hypothesis that loose coupling is always the best solution in distributed work. Questioning a hypothesis does not mean that the opposite is true; it simply raises new research questions about distributed work and encourages us to investigate coupling of work and, in particular, identify in more detail the differences and conditions for coupling of distributed work.

Finally, we found that the *organizational managerial aspects* in our cases point to the importance of creating enabling conditions for collaboration including contractual agreements and policies that ensure members of the global teams benefit when both sites benefit. This ensures that budget discussions do not interfere with the work and collaboration directly, but are kept at a different managerial level. There was a clear change in the GLOBALSOFT case when the organization went from outsourcing to

Table II. Revisiting the CSCW Fundamentals: Summary of Directions for Future Research

Common Ground —Continues to be a fundamental challenge in distributed collaboration —Future research should investigate common ground in terms of the dual agenda of • creating and developing shared language, and professional specific domain knowledge • identifying, creating, maintaining, and applying processes and methods that support global collaboration
Collaboration readiness —Continues to be a fundamental challenge in distributed collaboration —Future research should investigate collaboration readiness: • collaboration readiness should *not* be conceptualized as a singular continuum • but rather as a multiplicity of dimensions and relations that are not bidirectional by nature
Collaboration technologies readiness —Can no longer be identified as a fundamental challenge in distributed collaboration —Future research should instead investigate technology appropriation in terms of • the extra work required to create sociotechnical connections (e.g., relation work) • the design challenge of creating stability, accessibility, and availability of collaborative technologies in distributed work
Coupling of work —Loosely coupled work cannot be identified as a prerequisite for successful collaboration in distributed work. —Future research should investigate close coupling of work in more detail, in particular to identify the differences and conditions for coupling of work in distributed work related to • types of collaborative tasks • types of organizational structures
Organizational managerial aspects —Identifying organizational structures and managerial concerns is an essential concern for CSCW research on distributed work —Future research should investigate the organizational managerial aspects of distributed work in terms of • creating enabling conditions (including appropriation of collaborative technologies) for common ground, collaboration readiness, and closely coupled work • also taking into account the contractual agreements and policies related to inter- and intraorganizational structures

offshoring arrangements. Also, previous research has pointed to the issue of deception in distributed work [Bradner and Mark 2002]. Finally, rotation of employees both within local teams as well as across remote locations (on-site or offshore) was an important factor in terms of creating a large group of technical experts with experience in international collaboration and communication, thus reducing bottlenecks in global collaboration. Below Table II summarizes the findings.

6. CONCLUSION

We set out to analyze the distance framework from 2000 [Olson and Olson 2000] comprising the five concepts of common ground—collaboration readiness, collaboration technology readiness, coupling of work, and organizational management—by challenging our current theoretical understanding with the empirical data from our four ethnographic studies of global software development. In addition, we examined the organizational managerial practices of each case to determine which aspects of managerial practices were crucial for the success of the cases and thus might be suggested as possible best practices.

We found that common ground and collaboration readiness are still important analytical dimensions of distributed work and thus provide interesting design challenges for collaboration technology design and social practices. We also found that collaboration technology readiness is more an issue of organizational practices and unstable technology than the lack of knowledge about technologies. Moreover, we found that closely coupled work supports participants working remotely, which runs counter to previous findings. Our empirical observations suggest that the challenge for CSCW design of collaboration technologies concerns the challenges of bringing stability, availability, and accessibility to the collaboration technologies supporting frequent interaction across time and space. Finally, we found that managerial practices of finding ways to routinely rotate mediators or boundary spanners, who are critical to making the collaboration function well, are critical in retaining and building up pools of experts who can collaborate across geographical, temporal, and cultural distances. All four empirical cases were studied in real-life settings of global software development, and three of the cases were focused on collaboration between sites in Denmark directed at India (two cases) and Philippines (one case); the final case was focused on sites in India collaborating with sites in Europe (Germany and the United Kingdom). Thus, the majority of our data is based in either Denmark or India. The time difference between these sites is in all cases between 4 and 5 hours; thus, if we had studied cases in which the time difference was greater (such as East Coast United States and India, where there basically is no overlap), other pertinent findings might arise. Thus, future research is required to investigate the issue on time difference as a fundamental for distributed work.

We hope this article can be the foundation for new research on distance at a time when working remotely is commonplace and becoming less difficult.

ACKNOWLEDGMENTS

This study was conducted within the context of the interdisciplinary and interorganizational research project “Next Generation Technologies and Processes for Global Software Development,” #10-092313 a research project that is financially supported by the National Council for Strategic Research, Ministry of Science, Innovation and Higher Education in Denmark. For further information, see www.nexgsd.org. We would like to thank all companies that opened their doors and let us into their organizations to study their distributed collaboration.

Although Pernille Bjørn is the lead author on this paper, Morten Esbensen, Rasmus Eskild Jensen, and Stina Matthiesen each were the main drivers for conducting much of the empirical work presented in this paper. Others also participated in the data collection, and thus we would like to acknowledge their work. The empirical work for INIT was conducted in collaboration with Pernille Bjørn, Anne-Marie Søderberg, and S. Krishna. The empirical work for GLOBALSOFT was conducted in collaboration with Pernille Bjørn, Rasmus Eskild Jensen, Lars Rune Christensen, and Helle Storm. Although the empirical contact and study design for SCANDIABANK and INDK were created and negotiated by Pernille Bjørn, the empirical work was conducted by Stina Matthiesen and Lise Møller Petersen in SCANDIABANK, and by Morten Esbensen and Steven Jeuris in INDK.

We would also like to acknowledge and thank Judith Olson and Gary Olson for providing extensive feedback and comments on earlier drafts of this article while Pernille Bjørn was a visiting faculty member at UCI.

REFERENCES

G. Avram, L. Bannon, J. Bowers, and A. Sheehan. 2009. Bridging, patching and keeping the work flowing: Defect resolution in distributed software development. *Computer Supported Cooperative Work (CSCW): An International Journal* 18, 477–507.

E. Balka, P. Bjørn, et al. 2008. Steps towards a typology for health informatics. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'08)*. ACM.

P. Bjørn. 2003. Re-negotiating protocols: A way to integrate groupware in collaborative learning settings. In *Proceedings of the 11th European Conference on Information System*.

P. Bjørn and L. R. Christensen. 2011. Relation work: Creating socio-technical connections in global engineering. In *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'11)*. Kluwer Academic: 133–152.

P. Bjørn and M. Hertzum. 2006. Project-based collaborative learning: Negotiating leadership and commitment in virtual teams. In *Proceedings of the 5th Conference on Human Computer Interaction in Southern Africa (CHI-SA'06)*.

P. Bjørn and O. Ngwenyama. 2009. Virtual team collaboration: Building shared meaning, resolving breakdowns and creating translucence. *Information Systems Journal* 19, 3, 227–253.

P. Bjørn and O. Ngwenyama. 2010. Technology alignment: A new area in virtual team research. *IEEE Transactions on Professional Communication* 53, 4, 382–400.

P. Bjørn, A. Scupola, et al. 2006. Expanding technological frames towards mediated collaboration: groupware adoption in virtual learning teams. *Scandinavian Journal of Information Systems* 18, 2, 3–42.

A. Boden, G. Avram, et al. 2009. Knowledge management in distributed software development teams: Does culture matter? In *Proceedings of the International Conference on Global Software Engineering (ICGSE'09)*. IEEE, 18–27.

A. Boden, F. Rosswog, et al. 2014. Articulation spaces: Bridging the gap between formal and informal coordination. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work (CSCW'14)*.

N. Boulus and P. Bjørn. 2008. A cross-case analysis of technology-in-use practices: EPR-adaptation in Canada and Norway. *International Journal of Medical Informatics* 79, 6, 97–108.

E. Bradner and G. Mark. 2002. Why distance matters: Effects on cooperation, persuasion and deception. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'02)*.

L. Christensen and P. Bjørn. 2014. Documentscape: Intertextuallity, sequentiality and autonomy at work. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*.

L. R. Christensen, R. E. Jensen, et al. 2014. Relation work in collocated and distributed collaboration. In *Proceedings of the 11th International Conference on the Design of Cooperative Systems*.

H. Clark and S. Brennan. 1991. Grounding in communication. In *Perspectives on Social Shared Cognition*, L. Resnick, J. Levine, and S. Teasley, eds. American Psychological Association, Washington, DC, 127–149.

A. Crabtree, P. Tolmie, et al. 2013. How many bloody examples do you want? Fieldwork and generalisation. In *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'13)*.

C. D. Cramton. 2001. The mutual knowledge problem and its consequences for dispersed collaboration. *Organization Science* 12, 3, 346–371.

C. D. Cramton and P. Hinds. 2005. Subgroup dynamics in internationally distributed teams: Ethnocenterism or cross-national learning. *Research in Organizational Behavior* 26, 233–265.

K. M. Eisenhardt. 1989. Building theories from case study research. *The Academy of Management Review* 14, 4, 532–550.

T. Erickson and W. A. Kellogg. 2000. Social translucence: An approach to the designing systems that support social processes. *ACM Transactions on Computer-Human Interaction* 7, 1, 59–83.

M. Esbensen and P. Bjørn. 2014. Routine and standardization in global software development. In *GROUP*. ACM.

T. Finholt and L. S. Sproull. 1990. Electronic groups at work. *Organization Science* 1, 1, 41–64.

E. Gerson. 2008. Reach, bracket, and the limits of rationalized coordination: Some challenges for CSCW. In *Resources, Co-evolution and Artefacts: Theory in CSCW*, M. Ackerman, H. Christine, T. Erickson and W. A. Kellogg, eds. Springer, 193–220.

R. Grinter. 2003. Recomposition: Coordinating a web of software dependencies. *Computer Supported Cooperative Work (CSCW): An International Journal* 12, 297–327.

J. Grudin. 1994. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM* 37, 1, 92–105.

J. Grudin and R. Grinter. 1995. Ethnography and design. *Computer Supported Cooperative Work (CSCW): An International Journal* 3, 55–59.

C. Gutwin and S. Greenberg. 2002. A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work (CSCW): An International Journal* 11, 411–446.

C. Gutwin, R. Penner, et al. 2004. Group awareness in distributed software development. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'04)*. ACM.

R. Harper, J. Hughes, and D. Shapiro. 1989. Working in harmony: An examination of computer technology in air traffic control. In *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, J. Bowers and S. Benford, eds. Elsevier, North-Holland, Amsterdam.

C. Heath and P. Luff. 1992. Collaboration and control: Crisis management and multimedia technology in london underground line control rooms. *Computer Supported Cooperative Work (CSCW): An International Journal* 1, 69–94.

J. Herbsleb. 2007. Global software engineering: The future of socio-technical coordination. *Future of Software Engineering (FOSE'07).* IEEE Computer Society, Washington, DC.

J. Herbsleb and A. Mockus. 2003. An empirical study of speed and communication in globally-distributed software development. *IEEE Transactions on Software Engineering* 29, 3, 1–14.

J. Herbsleb, D. Paulish, et al. 2005. Global software development at Siemens: Experience from nine projects. In *Proceedings of the International Conference on Software Development (ICSE'05).* ACM.

J. Herbsled, A. Mockus, et al. 2000. Distance, dependencies, and delay in a global collaboration. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'00).* ACM, 319–328.

M. Hertzum and J. Pries-Heye. 2011. Is minimizing interaction a solution to cultural and materity inequality in offshore outsourcing. In *SourceIT: Balancing Sourcing and Innovation in Information System Development*, M. Hertzum and C. Jørgensen, eds. Tapir Academic Press, Tronheim, Norway.

P. Hinds and M. Mortensen. 2005. Understanding conflict in geographical distributed teams: The moderating effects of shared identity, shared context, and spontaneous communication. *Organization Science* 16, 3, 290–307.

S. L. Jarvenpaa and D. E. Leidner. 1999. Communication and trust in global virtual teams. *Organization Science* 10, 6, 791–815.

R. E. Jensen and P. Bjørn. 2012. Divergence and convergence in global software development: Cultural complextities as societal structures. In *COOP: Design of Cooperative Systems*. 123–136.

A. Majchrzak, R. E. Rice, et al. 2000. Technology adaptation: The case of a computer-supported interorganizational virtual team. *MIS Quarterly* 24, 4, 569–600.

A. Malhotra, A. Majchrzak, et al. 2001. Radical innovation without collocation: A case study of Boeing–Rocketdyne. *MIS Quarterly* 25, 2, 229–249.

G. Mark. 2001. Extreme collaboration. *Communications of the ACM* 45, 6, 90–93.

G. Mark. 2002. Conventions and commitment in distributed CSCW groups. *Computer Supported Cooperative Work (CSCW): An International Journal* 11, 349–387.

G. Mark, S. Abrams, et al. 2003. Group-to-group distance collaboration: Examining the space between. In *Proceeding of the 8th European Conference on Computer Supported Cooperative Work.* 99–118.

S. Matthiesen and P. Bjørn. 2015. Why replacing legacy systems is so hard in global software development: An information infrastructure perspective. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'15)*. ACM.

S. Matthiesen, P. Bjørn, et al. 2014. Figure out how to code with the hands of others: Recognizing cultural blind spots in global software development. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'14)*. ACM.

N. H. Møller and P. Bjørn. 2011. Layers in sorting practices: Sorting out patients with potential cancer. *Computer Supported Cooperative Work (CSCW): An International Journal* 20, 123–153.

B. Nardi. 2005. Beyond bandwidth: Dimensions of connections in interpersonal communication. *Computer Supported Cooperative Work (CSCW): An International Journal* 14, 91–130.

G. M. Olson and J. S. Olson. 2000. Distance matters. *Human-Computer Interaction* 15, 139–178.

J. Olson, E. Hofer, et al. 2008. A theory of remote scientific collaboration. In *Scientific Collaboration on the Internet,* G. Olson, A. Zimmerman, and N. Bos, eds. MIT Press, Cambridge, MA.

J. Olson and G. Olson. 2014. *Working Together Apart: Collaboration over the Internet.* Morgan & Claypool.

W. Orlikowski. 1992. Learning from notes: Organizational issues in groupware implementation. In *Proceedings of the ACM Conference on Conference on Computer Supported Cooperative Work (CSCW'92).* ACM, New York.

W. J. Orlikowski and D. C. Gash. 1994. Technological frames: Making sense of information technology in organizations. *ACM Transactions on Information Systems* 12, 2, 174–207.

N. Pollock and R. Williams. 2010. e-intrastructures: How do we know and understand them? Strategic ethnography and the biography of artefacts. *Computer Supported Cooperative Work (CSCW): An International Journal* 19, 6, 521–556.

D. Randall, R. Harper, et al. 2007. *Fieldwork for Design: Theory and Practice*. Springer, London.

K. Schmidt. 2002. The problem with "Awareness." *Computer Supported Cooperative Work (CSCW): An International Journal* 11, 285–298.

K. Schmidt. 2009. Divided by a common acronym: On the fragmentation of CSCW. In *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'09)*. Springer.

K. Schmidt and L. Bannon. 1992. Taking CSCW seriously: Supporting articulation work. *Computer Supported Cooperative Work (CSCW): An International Journal* 1, 1–2, 7–40.

K. Schmidt and L. Bannon. 2013. Constructing CSCW: The first quarter century. *Computer Supported Cooperative Work (CSCW): An International Journal* 22, 345–372.

K. Schmidt, I. Wagner, et al. 2007. Permutations of cooperative work practices: A study of two oncology clinics. In *GROUP*. ACM.

A.-M. Søderberg, S. Krishna, et al. 2013. Global software development: commitment, trust and cultural sensitivity in strategic partnerships. *Journal of International Management* 19, 4, 347–361.

A. Strauss. 1988. The articulation of project work: An organizational process. *The Sociological Quarterly* 29, 2, 163–178.

A. Strauss, S. Fagerhaugh, et al. 1985. *Social Organization of Medical Work*. The University of Chicago Press, Chicago.

S. Teasley, L. Covi, et al. 2000. How does radical collocation help a team succeed? In *Proceedings of the ACM Conference on Computer Computer Supported Cooperative Work (CSCW'00)*. ACM, 339–346.

M. Tyre and W. Orlikowski. 1994. Windows of opportunities: Temporal patterns of technological adaptation in organizations. *Organization Science* 5, 1, 98–118.

A. Voida, J. Olson, et al. 2013. Turbulence in the clouds: Challenges of cloud-based information work. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*. ACM.

Received May 2014; revised September 2014; accepted September 2014