A conceptual framework to study the role of communication through social software for coordination in…

Article *in* Information and Software Technology · March 2015

DOI: 10.1016/j.infsof.2015.02.013

2 authors:

Rosalba Giuffrida
IT University of Copenhagen
10 PUBLICATIONS 59 CITATIONS

SEE PROFILE

Yvonne Dittrich
IT University of Copenhagen
146 PUBLICATIONS 1,145 CITATIONS

SEE PROFILE

A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams

Rosalba Giuffrida

IT University of Copenhagen — Software and System Section
Copenhagen, Denmark rogi@itu.dk

Yvonne Dittrich

IT University of Copenhagen — Software and System Section
Copenhagen, Denmark ydi@itu.dk

Abstract

Background: In Global Software Development (GSD) the lack of face-to-face communication is a major challenge and effective computer-mediated practices are necessary to mitigate the effect of physical distance. Communication through Social Software (SoSo) supports team coordination, helping to deal with geographical distance; however, in Software Engineering literature, there is a lack of suitable theoretical concepts to analyze and describe everyday practices of globally-distributed software development teams and to study the role of communication through SoSo.
Objective: The paper proposes a theoretical framework for analyzing how communicative and coordinative practices are constituted and maintained in globally-distributed teams.
Method: The framework is based on the concepts of communicative genres and coordination mechanisms; it is motivated and explicated through examples from two qualitative empirical cases.
Results: Coordination mechanisms and communicative genres mutually support each other. In particular, communication through SoSo supports team members in establishing, developing and maintaining social protocols within the distributed team. Software Engineering tools and methods provide templates for coordination mechanism that need to be adapted and adopted in order to support the project at hand. SoSo serves as a medium for the necessary metawork. The theoretical framework proposed is used to describe both the practices in an established industrial project and the establishing of practices in three student teams. The framework allows explaining the heterogeneity of practices observed.
Conclusions: This paper presents a conceptual framework to study the role of communication through SoSo for coordination in GSD. The usefulness of the framework is supported by empirical findings on the role of SoSo. The theoretical framework can be beneficial for future research that aims to analyze and describe not only the role of SoSo, but also how communicative and coordinative practices can be established and maintained in GSD teams.

Keywords: Global Software Development, Communicative Genres, Coordination Mechanisms, Social Software, Human Aspects

1. Introduction

Software Engineering (SE) is cooperative work [1], and software developers must coordinate their individual activities with tasks performed by other team members [2] in their everyday practices. Coordination relies on communication, direct communication, as well as communication mediated by code, documentation and artifacts. Communication is fundamental not only to coordinating the cooperative work, but also to establishing and to maintaining effective coordination mechanisms [3]. In Global Software Development (GSD) settings, effective coordination is challenging [4] due to the lack of face-to-face communication [5] between distributed team members. On the one hand, research in GSD aims to overcome this challenge by improving processes and tools for supporting cooperation in distributed teams, e.g. reducing intensive collaboration [5], increasing formal documentation [6], and working on organizational factors such as processes, structure, and goal alignment [7]. On the other hand, GSD research shows the success of agile processes in GSD settings [8] that depend on close collaborations and frequent informal face-to-face communication, rather than lengthy documentation. These premises motivate the necessity of further studies on tools and practices in GSD, as no standard recommendations are as yet available in the field.

While the main media for communication in distributed teams have traditionally been email, phone, and video conferencing systems, nowadays communication also takes place in the so-called Social Software (SoSo). SoSo is often referred to as “social media”, “web 2.0”, “user generated content” by practitioners and researchers. Kaplan and Haenlein [9] combine the different terminologies defining SoSo as “a group of Internet-based applications, built on the ideological and technological foundations of Web 2.0, that allow the creation and exchange of User Generated Content (UGC).” Essentially, SoSo encompasses a range of software systems that allow users to interact and share information, such as: Instant Messaging (IM), Internet Forums, Blogs, Microblogs, Wikis, Social Network Sites, and Social Bookmarking. Very little research focuses on the usage of SoSo in GSD, as highlighted in a Systematic Mapping Study performed by the authors of this paper [10] [1]. The use of SoSo in GSD practices might, however, be more wide spread than what is visible in the actual research publications [10]. Some indications are provided by the success of many open-source projects (OSS) that are often globally distributed and mainly coordinated through the wide use of SoSo, such as Wikis, Forums and Instant Messaging [11]. The real challenge lies in answering the question about why and how SoSo can provide useful communication channels for distributed collaboration; in other words, how it is used in the everyday practices of distributed software teams.

Traditional Software Engineering (SE) as a discipline is “mainly concerned with the formal principles, the technical basis and the methodological support for software development, rather than the reflection of software practice as a human activity that goes beyond the engineering framework” [12]. Often, practices in situated action [13] differ from pre-defined SE methods and processes. Detailed descriptions of practices provide the basis for understanding the social factors [12] that influence effective practices [14], as well as for designing better tools [15]. Practice-based approach studies are becoming more widespread in SE research, e.g., [14], [16] as well as in GSD research, e.g., [17], [18], compared to other approaches used to inform tool design, such as the cognitive approach, widely used in human-computer interaction to study the cognitive processes of programmers, i.e. the mental processes involved in programming [19]. A practice-based approach [20] is adopted by several researchers to illustrate GSD work practices: Sigfridsson and Sheehan [17] describe an Open Source community called PyPy, Boden et al. [21] study the coordination practices in distributed software development of small enterprises, and Avram [15] investigates a project with the purpose of better understanding collaborative work and knowledge management processes in distributed software development settings. These research studies all describe work practices of software engineers; however, they suffer in proposing an effective way of analyzing such practices.

Computer-supported Cooperative Work (CSCW) tradition offers many theoretical concepts that can be used to analyze and describe practices in SE as well as in GSD settings. In particular, in GSD most of the cooperation is computer-mediated, thus relying heavily on the use of artifacts, both for coordination and for communication. In particular, most of the GSD artifacts are traceable, and researchers can analyze them to investigate the actual practices. Coordination artifacts in GSD consist of, e.g., bug reports [2], documentation, files or source code, while communication artifacts can be, e.g., comments in the source code, descriptions provided while committing the code in versioning systems, email or SoSo. In CSCW tradition, the facilitating role of artifacts in collaboration is well recognized. Thus, it appears promising to find suitable theoretical concepts from the CSCW tradition to investigate the use of artifacts for communication and collaboration in GSD settings.

This paper presents a novel conceptual framework for analyzing and describing the role of SoSo in GSD everyday practices, motivated and explicated through examples taken from two empirical cases. The empirical cases are described in detail in two previous articles [22] [23]. The purpose of this paper is to develop and illustrate the theoretical framework, showing its applicability and flexibility in different situations. In particular, both established and establishing distributed teams have been employed to show the establishing and the development of common practices in GSD settings. For the authors of the present paper, a framework is a set of related concepts that is used to analyze cases and to explain phenomena: by relating concepts together in a single framework, it is possible to describe not only one specific case but several cases of different natures. Thus, a framework allows generalizing from individual empirical cases to general phenomena. The framework proposed is based on the theoretical concepts of communicative genres [24] and coordination mechanisms [3], both based on the notion of social protocols, a set of rules, conventions, and policies shared by people involved in the cooperative activity [3]. The idea behind the framework is that the development of common social protocols is crucial for achieving effective communicative and coordinative practices, as well as for the adaptation of models and methods. The framework shows the relationship between the concepts of coordination mechanisms and communicative genres both during the establishment and the implementation of common coordination and communication practices.

The analysis of communicative and coordinative practices of the two cases shows that SoSo is especially useful in establishing, developing and maintaining social protocols. In this context, the role of SoSo is to support communication and its key function is complementing collaborative SE tools that provide templates for coordination mechanisms. The theoretical framework allows for both describing practices in an established industrial project [22] and analyzing the development of practices in three establishing student teams [23], permitting the explaining of the heterogeneity of practices observed and showing how dynamics develop and evolve in the different teams. The framework proposed can be beneficial for future research that aims not only to analyze and describe the role of SoSo, but also to show how communicative and coordinative practices in general are established and maintained in globally-distributed teams.

The reminder of the paper is organized as follows. The next section summarizes related work on coordination and commu-

[1] Please note that the search string used in the Systematic Mapping Study is built to include a wide spectrum of research paper: the different terms used by practitioners and researchers to refer to SoSo have been employed — such as social media, social software, web 2.0, user generated media, user generated content, instant messaging, wiki, social network, social bookmark, blog, microblog, social tagging, facebook, twitter — and they have been combined with terms to identify: (a) teams which are not co-located; (b) SE and its main phases; and (c) known purposes for using SoSo. Moreover, the snowballing technique has been adopted to extend the amount of research paper retrieved.

nication in Software Engineering. Section 3 describes the two empirical cases used for explicating the theoretical framework proposed in this paper and Section 4 explicates the research methodology. Examples taken from the two empirical cases are used in Section 5 to describe in greater detail the concepts that constitute the theoretical framework, and in Section 6 to highlight the relation between the concepts. Finally, Section 7 aggregates the theoretical concepts in a unitary theoretical framework for communicative and coordinative practices. Section 8 discusses the framework, highlights contributions and implications, and in brief, reports the limitations of the study. Final conclusions are presented in Section 9.

2. Related Work

Global Software Development (GSD) is increasingly becoming common practice in the software industry [25]. GSD means splitting the development of the same product or service between globally-distributed sites [11]. There are many potential benefits that can arise from GSD: lower development costs due to salary savings, decreased development time due to time-zone effectiveness, reduced time to market, and access to the most talented developers [25]. Developing software as a team is a challenging task, but developing software as a globally-distributed team is even more challenging due to geographical distance [6]. In particular, main challenges arise in communication and coordination due to the lack of face-to-face communication between distributed team members.

Communication through SoSo has been promoted to support distributed teams, helping to overcome geographical distance. Several examples are reported in [10]: Instant Messaging (IM) can replace planned or impromptu face-to-face meetings that are not feasible in GSD settings [27] [28], [29]; Wikis can be used for knowledge sharing [30]; microblogs can generate virtual "water-cooler" conversations and can be used as an informal communication channel [31]; and tagging can support software development [32] and help to bridge the gap between social and technical aspects in software development [33]. Very little research focuses on the usage of SoSo in GSD, and no theoretical concepts are used to analyze the role of SoSo in GSD practices[10]. Communication through SoSo supports collaboration in distributed teams, and plays a special role in establishing, fostering and maintaining coordination mechanisms.

Research on coordination encompasses a variety of disciplines such as organizational theory, information systems, and sociology of work [34]. Coordination is defined in the Coordination Theory proposed by Malone and Crowston [35] as "managing dependencies between activities"; in this context, coordination is achieved by one or more "coordination mechanisms," each addressing one or more dependencies in a situation. Coordination mechanisms are also defined as "the organizational arrangements that allow individuals to realize a collective performance" [34]. In the literature, these organizational arrangements often involve tools, technologies, or interactions that bring interdependent elements together [34]. In SE as well as in GSD literature, the concept of "coordination mechanism" is widely used to indicate a mix of a broad set of practices, methods, processes, and tools, as the "mechanism" concept is frequently used as a general term not related to actual SE practices [4, 36, 37]. For example, in the context of an agile software team [38], tools such as wiki, activities such as the daily standup meeting, roles such as the project manager, and artifacts such as the product backlog, are all considered coordination mechanisms. Similarly, in the context of a GSD project [7], a set of coordination mechanisms of various natures are analyzed, such as centralized team structure, documentation, periodic commit, communication tools and periodic meetings. Empirical evidence provided by Schmidt and Simone [3] in the context of Computer-supported Cooperative Work (CSCW) research, however, shows the widespread use of coordinative practices that rely on coordinative artifacts. Thus, as part of the actual CSCW practices, and based on the use of artifacts for coordination purposes, "coordination mechanisms" are defined by Schmidt and Simone [3] with a rigorous definition that encompasses concepts such as social protocols and articulation work, that appear promising to understand coordinative practices in GSD, using a theoretical concept that allows to explain not only how different practices occur in different projects but also to compare them.

McChesney and Gallagher [39] propose a framework to describe coordination activities in SE. They argue that "there is a gap between existing process-oriented method for describing software processes and the situated activities in which software engineers engage when developing systems"; thus, a better understanding of actual coordination practices is necessary to bridge this gap. Following McChesney and Gallagher's approach, this paper describes communicative and coordinative practices in situated action [13] and shows how methods and processes are adapted by team members in these practices. In their article, McChesney and Gallagher relate coordination theory [35] and communication genres [24], using the framework to explain and interpret the complex web of personal interactions observed in two real-word software projects. Though McChesney and Gallagher attempt to analyze both coordinative and communicative practices in SE, they stick to coordinative practices, leaving the relationship between coordination and communication unexplored. They quickly abandon the concept of "coordination mechanisms" proposed by Schmidt and Simone [3] and replace it with the "communicative genres" notion by Yates and Orlokowski [24]. In our field material, "coordination mechanisms" and "communicative genres" appear as complementary concepts, both necessary to understand distributed software practices; as compatible concepts, both are based on the notion of social protocols. Thus, an elaboration of a more comprehensive framework appears necessary. In the framework proposed in this paper, the concept of "communicative genres" notion by Yates and Orlokowski [24] is used in combination with the concept of "coordination mechanisms" proposed by Schmidt and Simone [3]. In this paper, the two concepts are explicated and extended through several examples taken from two empirical cases. A detailed description of the two cases follows.

3. Description of the Cases

The theoretical framework proposed in this paper is developed based on two empirical cases: DHI — an industrial case — and the case of three student projects. An overview of the four distributed teams is presented in Table 1. In this paper, we use the terms *globally-distributed team* and *distributed team* interchangeably. Following Šmite et al.'s definition [26], in a distributed team, collaboration happens *intra-organization*, where team members are located in different locations (*offshoring*) and in different countries (*international*) through internal projects (*insourcing*). A distributed team consists of two or more sub-teams located in different sites with each sub-team comprising one or more individuals. According to this definition, in a distributed team, collaboration can occur in each site (*intra-sub-team*) and within the entire team (*intra-team* and *inter-sub-teams*). In this work, we focus on the intra-team collaboration that is generally computer-mediated, i.e., tracked in digital artifacts. Therefore, in the following, we describe in detail the four distributed teams under study and the tools used for the collaboration within the teams.

3.1. An Industrial Case: DHI

DHI is an independent international consulting and research organization. The company develops and uses high-end hydraulic simulation software. We have observed part of the World Bank Project (WB-Project), which has a considerable amount of software development. This is a globally-distributed project: five members are settled in Copenhagen, Denmark; seven members in Delhi, India, and one Project Area Manager (PAM) in Portland, USA. Danish members are experts with long-term software engineering backgrounds and hydraulic engineering competences; i.e., they are both domain experts of hydraulic engineering and software developers. Each PAM is responsible for one specific part of the software, collaborating with one or two Indian developers. Danish members take care of the management of the project, requirement specifications, quality assurance processes, as well as the design of the software and its implementation. Indian developers have several years of experience with software engineering, while the Indian team leader is a senior developer; the Indian sub-team is settled in DHI offices in Delhi. In greater detail, the five Danish team members are spread in four two-person-offices in two different floors. They have daily meetings at 10 a.m. CET in the office of the project manager to plan future activities, prioritize tasks, and assign incidents and tasks to team members. The meeting usually lasts about one hour. Often Danish team members visit each other's offices to have a brief chat. The Indian team members work in an open plan office. A big table is shared by all team members sitting on the two long sides of the table. The table is split into personal workspaces by small dividers. Sometimes the developers move to another workspace to communicate with other co-located team members. Formal co-located meetings in the Indian site were not observed by the researcher. Meetings involving the whole distributed team are rare, while collaboration across sites between PAMs and developers working on the same part of the project take place daily during the four overlapping working hours for different purposes: clarifying requirements, debugging pieces of software, and coordinating work tasks. The team follows an iterative process where PAMs are responsible for deciding, prioritizing as well as assigning tasks and requirements, while developers and testers are involved in the subsequent phase of the development process. An example of the process is described in Section 5.1 that describes the Incident Workflow.

We observed the team while working on the development of a Decision Support System (DSS) for water management in the Nile Basin. The observations took place during the final part of the development process of the first release of the system mainly during the testing phase of the same release. At that time, the whole team had worked together for one year on the project and some of the co-located team members had previously worked together on other projects. The project was successful, and the team has subsequently developed a second release of the software.

The main tool used by the team is Spira[2], a Test Management tool. The team though uses it to manage all development activities through the issue tracking system: the descriptions of features to develop, reports of incidents, assignments of tasks, and description of test cases. Spira automatically assigns an "incident number" to all defects, test cases and requirements. Skype[3] is the team's main tool for communication and team members are supposed to be available on Skype when they are at work. Team members use different Skype channels for synchronous or nearly synchronous collaboration: written IM, audio, and screen sharing. Usually, one-to-one Instant Messaging (IM) chats take place through Skype. For a detailed description of the DHI case, see the previous article that describes practices observed in the team [22].

3.2. The three Student Projects Case

The three projects under study are part of a GSD student cluster in collaboration with IT University of Copenhagen (ITU), Peking University (PKU), and Universidade Federal de Pernambuco (UFPE). The collaboration took place from February 2011 to May 2011. Students involved in the project were part of a Master's degree in Software Development, and the student cluster provided 15 ECTS [4] to each student. An academic supervisor provided the description of the product to be developed and evaluated the work performed by the students based not only on the code developed, but also on a final report produced by team members in each location and on an oral exam. Students' responsibilities comprised the system design, the requirement specifications, the development of the product, and the organization of the collaboration.

The teams "self-organized" their work, sharing roles, responsibilities, and decision taking. As a student project, team members did not work on a daily basis on the product and no fixed working hours were imposed. The teams, however, defined two

[2] http://www.inflectra.com/HomePage.aspx

[3] http://www.skype.com/

[4] http://ec.europa.eu/education/tools/ects_en.htm

Table 1: Overview of the teams.

Team	Time Zone	Team Members	Observations	Tools used
DHI team	Copenhagen GMT +1 New Delhi GMT +5:30 Portland GMT -7	1 project manager and 4 PAMs in Denmark 5 developers, 1 team leader and 1 tester in India 1 PAM in USA	During Fall 2010 4 months in Denmark and 2 weeks in India.	SpiraTeam, Skype, email.
Team A	Copenhagen GMT +1 Recife GMT -3	4 Denmark 6 Brazil	From February 2011 to May 2011. 3 months in Denmark and 1 week in India.	Assembla (in particular Forum and Wiki), phone, Skype (personal and group chat), email (rarely), and videoconference system.
Team B	Copenhagen GMT +1 Beijing GMT +8	5 Denmark 4 Chinese		
Team C	Copenhagen GMT +1 Beijing GMT +8	5 Denmark 3 Chinese		

fixed days a week for working on the project; due to the time difference and the different course schedules in each location, there were not many overlapping working hours between team members. In essence, synchronous communication took place only during the one-hour weekly meeting. Students were working on the project in meeting rooms or classrooms available in each University. The weekly meeting was carried on with remote team members in a video conference meeting room to coordinate the work of the whole team. Though the product developed was relatively small — it required only three months part time development — the students were asked to act as professionals, dealing with GSD challenges on their own. Some of the students were also part-time software developers in industrial settings. The commitment and the responsibility of the students to accomplish their tasks were not dissimilar to novice professional developers in software development. The situation quite resembled the case of starting up a new globally-distributed team, comparable somewhat with a small industrial team facing the same challenge.

Tools used by the three distributed teams were: Skype, for video conference and IM chats, emails and Assembla[5], an issue tracker system with additional functionalities such as file sharing, Wiki, Forum and an integration to Subversion (SVN) [6], the version control system used. Most of the collaboration took place in Assembla and during the weekly meeting of about one hour through Skype. Assembla was used for keeping track of the status of the project, for managing issues, for defining deadlines and as a shared repository for file exchange. Emails were rarely used: the forum of Assembla substituted the need for use of email for communication between team members: it worked as a common repository of the messages exchanged. Moreover, it was possible to enable the notification via mail functionality that updated team members via mail about the new events of Assembla. IM chats were rarely used due to the lack of overlapping working hours. A Skype group chat was used during the whole project by Team C; mostly used in the initial phases of the project. All teams succeeded in having a working prototype. The collaboration is considered successful based on the self-reported impressions of team members in the final report as well as on the observations by the researcher. A description of the three teams follows.

3.2.1. Team A

The goal of the project was to design, develop, and deploy web service robots that could automatically surf different web pages and web services, extract and submit information, as well as manage and coordinate various tasks. The weekly meeting took place every Friday at 11 a.m. Brazilian time, i.e., 3 p.m. Danish time. Both ITU and UFPE offered a video conference room fully equipped with an advanced system that allowed high quality video and audio conference. However, often team members experienced issues in oral English and they introduced a mute-time, in which the microphone of each site was muted on the meetings in order to accommodate for internal discussions and for those who needed translation into their own language. The weekly meetings and Assembla, a common repository that could broadcast information to all team members, were apparently sufficient for the collaboration. However, the team experienced problems at the end of the project integrating the two parts of the code. The Danish team members reported that the collaboration was unsuccessful, despite having a running product and gaining good grades. Since limited communication happened during the project, the distributed team did not manage to exploit global collaboration; Thus, working globally resulted in the project requiring greater integration.

3.2.2. Team B

The goal of the project was to develop a context-aware system that would assist passengers navigating around an airport. The weekly meeting was planned every Thursday at 9 a.m., Danish time, i.e., 4 p.m., Chinese time. During the weekly meeting, team members experienced many challenges in using Skype or QQ [7] for video conferencing, and thus they often needed to use a normal phone to communicate. Even with the use of the landline, the quality of the audio was often poor and the accents were very strong, with team members not being able to understand each other. Due to the lack of a visual channel, it was not always clear what was happening in the remote site when responses were not heard or understood. Furthermore, ITU team members were often puzzled when Chinese members

[5] `http://www.assembla.com`

[6] `http://subversion.apache.org/`

[7] `http://www.imqq.com`

were talking in Chinese, without informing the Danish members about the reason. [8] During the design phase of the project, it was decided how to distribute the work among the nine team members, sharing roles and responsibilities and having people responsible for each part of the project in both locations. The collaboration had several challenges, especially at the beginning, which brought frustration to team members. However, over time, team members managed to deal with the difficulties in communication and collaborated more effectively, and thus became effective in the final phase.

3.2.3. Team C

The goal of the project was to design and implement an e-collaboration tool designed to support the collaboration in a global project. Weekly meetings took place on Friday at 1 pm Danish time, i.e., 8 pm Chinese time. Skype and QQ resulted in very poor quality for video conferencing and thus the meetings were held on with the use of a normal phone line for the audio; however, a visual channel was kept through Skype with audio muted. The team adopted Scrum as a development model and used it rather rigorously, having weekly standup meetings. Standup meetings were locally performed and video recorded, and then shared with remote team members. Videos were watched by remote team members. Team members were thus kept up-to-date on the status of the project, and they became more familiar with each other (voice, face, accent, ...), as well as with the environment (the videos were meant also to show remote members the environment in which each sub-team was working). Each sub-team started working on different ends of the system (the ITU sub-team working on front-end and database, the PKU sub-team working on the back-end). The two sub-teams then combined the work in the intermediate layer (engine) of the solution without encountering major difficulties. The collaboration was considered to be successful because, despite spatial and temporal distance, team members managed to have smooth collaboration, without major frustration.

4. Method

This paper explicates the novel conceptual framework through examples taken from two ethnographically-inspired studies[9]: an industrial case and three student projects case, as described in the previous section. The industrial case was used to observe the well-established practices of practitioners in an established team, while the three student projects were used to compare the establishing and maintenance of practices in newly formed distributed teams. The student case was meant to simulate a starting up GSD project; students were thus acting as novice professionals. The two cases have been selected to show that both establishing, and established, teams can be used in the framework.

4.1. Data Gathering

The two real world cases were observed and analyzed through different data collection techniques: non-participant observations, semi-structured interviews, informal in-situ interviews, and document analysis. In order to carefully keep track of the investigation, a research diary was kept to observe each project. Meetings and interviews were taped and transcribed; the content of computer-mediated communication through SoSo was analyzed both qualitatively and quantitatively, using interaction analysis [40]. By using non-participant observations, where the first author, as a researcher, observed the daily routines of the teams but did not actively work on the development of the product, it was possible to observe how collaboration took place between team members in the different projects. Semi-structured interviews gave the possibility of clarifying uncertainties and asking details about specific issues; these were used mostly for triangulation purposes and were not coded in detail. Semi-structured interviews were specifically designed for the interviewees of each project; an example of an interview guide used in the DHI case is reported in Appendix A.

Interaction analysis [40] was performed for analyzing SoSo logs: interaction analysis (from a sociolinguistic perspective) is concerned with the structure of social interaction manifested in the conversation, emphasizing features of social context. This technique was used for analyzing the exchange of messages between team members, and bottom-up dimensions were applied as codes in the analysis of SoSo logs: in particular, IM logs for the DHI case, and Forum and Wiki logs for the student case. Results of the log analysis were triangulated with the examination of other documents, such as email, technical documents and logs of collaborative tools, compared with collected field notes that were subsequently verified through semi-structured interviews with team members. A final workshop was organized in the industrial case to summarize the outcomes and to support researchers and practitioners in reflecting together on the findings obtained. Similar feedback was not provided for the student participants, as the project was closed during the analysis of the field material.

By using multiple ways of collecting data and combining different kinds of methods, it was possible to triangulate the findings [41]. Different kinds of triangulation were adopted to assure the trustworthiness of the research and its results during the whole analysis of the specific cases: results of the log analysis were checked with researcher's observations, with relevant documentation and artifacts provided to the researchers, and with members checking through informal in-situ interviews, semi-structured interviews, and workshops.

4.2. DHI Case

The first author collected field material and observed the distributed team for a total of 30 working days spread out over four months from September 2010 until January 2011. She visited

[8] During an on-site visit, the first author discovered that Chinese members were clarifying what was just said by Danish members and were trying to formulate a proper reply, dealing with their uncertainty about their English skills.

[9] The approach adopted is ethnographically-inspired as it does not provide an ethnography as a result, observations have been performed for a limited amount of time, and the researcher was not involved in the project, as a non-participant observer.

the Copenhagen site for most of the time and the Delhi site for two weeks at the beginning of November 2011. The researcher was placed in the office of the project manager in Copenhagen for most of the time; however, she also spent seven days of observations in the office of other PAMs in Copenhagen and ten days in the open space with Indian developers. Unfortunately, it was not possible to visit the team member in Portland due to time and cost constraints. However, he spent a week in Copenhagen, where he and the researcher sat in the same office for two days.

The team provided the researcher with free access to Spira, to the IM chats that occurred between sites during the days of observations, and to some email. The 30 days of IM chats resulted in a total of 104 pages of a Word document that were initially analyzed and coded. Informal interviews were held during the observations with most of the team members, both in the Danish site and the Indian site in order to gain more insights into the practices observed and to verify preliminary findings. However, two semi-structured interviews were also performed: one reported as an example in Appendix A with the project manager after the researcher had visited the Indian site, and another with the member located in Portland after the end of the project. Interviews were used to further elaborate and triangulate the findings, and to gain more insights that were not visible from observations and log analysis.

The analysis of the data was initially performed during the observation period through several iterations of analysis of the IM logs and observations. Through these iterations, it was possible to reflect on the observed practices and to re-adapt the data collection. Once the observation period ended, researchers re-analyzed the field material in order to summarize the findings. A further analysis of a sample of the IM chats and part of the artifacts was performed subsequently in order to support the development of the theoretical framework and the use of the theoretical concepts of coordination mechanisms and communicative genres that were not used in the previous analysis. The second analysis was interrupted when no new genres and coordination mechanism were identified; thus, the saturation criteria of qualitative research was met.

4.3. Student Case

The first author attended most of the meetings from the Danish site during the three months of collaboration and the one-week visit to the PKU site, during which time she collected pictures, took notes during observations of meetings, and informally interviewed one or two participants in each location. Unfortunately, it was not possible to visit the Brazilian site due to time and cost constraints. The teams provided the researchers with free access to the whole Assembla repository, including forum discussion, documentation, wiki, files exchanged, and source code. Moreover, the researchers had access to the Dropbox folder used by Team A, to the group discussion carried on in Skype by Team B, to some emails, and to some one-to-one IM chats between team members. In addition, a final report of the project was produced by the three Danish sub-teams and the Brazilian sub-team, and was provided to researchers.

Most of the communication between team members happened through the Assembla Forum and through the weekly meetings. The interaction analysis was performed after the end of the project on the whole Assembla Forum and on all Wiki pages; nevertheless, data from Skype chat, mail, and other Assembla sources were used as well to support evidence provided by the Forum analysis for the triangulation of the findings. The analysis of the data was performed while developing the conceptual framework based on the notions of coordination mechanisms and communicative genres, as well as after the project ended.

In particular, the first author coded all conversations carried on in the Forum and the Wiki pages of Assembla, identifying communicative genres. She incrementally developed the coding schema while coding the messages. The coding schema was then discussed with the second author. Moreover, in order to limit research biases, a colleague not involved in the project was asked to check the coding schema and to independently code a sample of the Forum messages and of the Wiki pages. Divergences in the coding were discussed and resolved. The coding schema is reported in Appendix B. Findings were triangulated with first author's observations, artifacts and documentation available in the Assembla repository, and with informal interviews with team members performed during the project in both site, with a semi-structured interview with one of the Chinese students and with a post-mortem semi-structured interview with two Danish team members.

4.4. Data Analysis

Previous research of the authors reported the analysis of the role of Instant Messaging (IM) in the DHI case [22] and the analysis of communicative and coordinative practices in Team C of the student case [23]. The theoretical framework presented in this paper has been developed based on the field material offered by both empirical cases. In the first paper [22], an interaction analysis [40] of the IM chats was performed based on their communicative purpose, and no theoretical concepts were adopted; in the second paper [23] the concepts of communicative genres and coordination mechanisms were used, showing a promising relation between them, one that deserves further investigation. Therefore, in this paper, the relationship between communicative genres and coordination mechanisms is further detailed, using the theoretical concepts to re-analyze part of the field material of the DHI case and to extend the analysis of the student team material with the two teams not studied in [23]. Coordination mechanisms in both cases were identified and analyzed, focusing on the investigation of the relation with communicative genres. Communicative genres were identified and coded in all Forum messages of the student case and in a sample of the IM chat of the DHI case. The coding schema of the communicative genres, reported in Appendix B, was developed while coding the whole SoSo communication in the student case in [23]. The communicative genres were also used to categorize the forum messages of the two student projects as well as a sample of the IM logs of the DHI case [22]. A detailed description of all practices observed in both cases is outside the scope of this paper. However, the theoretical framework is used for describing and analyzing exemplary practices observed, and it is motivated and explicated by the empirical material. The trustworthiness of the analysis is provided by (a) the member checking through semi-structured interviews and the final workshop in the industrial case, (b) case specific triangulation, and (c) cross-case triangulation. The next section explains the concepts used for the theoretical framework; through examples from the field material, the section brings the reader to the development of the theoretical framework, illustrated in Section 7.

5. Developing the Theoretical Concepts for the Framework

This section describes in detail the concepts of *coordination mechanisms* and *communicative genres* that are the basis for the theoretical framework presented in Section 7 and that are illustrated with examples taken from the empirical cases presented in Section 3. Building blocks of the two concepts are reinterpreted from the original definitions to provide a comprehensive understanding of the relationship between communicative and coordinative practices. Thus, the definitions of *communicative purpose*, *articulation work*, *social protocols* and *genre repertoire* are extended and discussed in detail in the second part of this section, providing additional examples.

5.1. Coordination Mechanisms

5.1.1. Definition

The concept of *coordination mechanisms* is defined by Schmidt and Simone [3] as follows:

> "A *coordination mechanism* is a specific organizational construct, consisting of a *coordinative protocol* imprinted upon a *distinct artifact* which, . . . , *stipulates* and *mediates* the articulation of cooperative work so as to reduce the complexity of *articulation work* . . . " [3] (emphasis in original)

The coordinative protocol consists of a set of rules, e.g., taken-for-granted ways of proceedings, established conventions, official policies, and standard operating procedures. The coordinative artifact is a "stable data structure expressed in a standardized graphical format" [3]. Schmidt and Simone [3] report, for example, about a bug report form[10], a two page form (the artifact) with several fields filled by different actors, which follows a set of agreed procedures and conventions (the protocol) and which stipulates the responsibilities to the different roles, the possible classifications of bugs, reports of bugs corrected, etc. The artifact is "the distinct and persistent symbolic construct in which the protocol is imprinted and objectified" [3]. In cooperative work settings characterized by complex task interdependence, "coordination mechanisms reduce the complexity of articulation work and alleviate the need for ad-hoc deliberation and negotiation" [3]. Since SE is cooperative work [1], the concept of coordination mechanism can be beneficial when studying GSD coordinative practices.

5.1.2. Extending the definition

The concept of coordination mechanism implicitly comprises the notion of purpose that is explicitly introduced in our theoretical framework. The purpose of the coordination mechanism is indicated in the name of the coordination mechanism itself. For example, an issue tracker mechanism allows the tracking of issues, while a file sharing mechanism has the purpose of sharing files. Further examples of coordination mechanisms are described in the following.

5.1.3. Examples

Many coordination mechanisms are used by software teams when developing software. Anytime an artifact is used to coordinate the cooperative work and team members agree on a social protocol about how to use it, an effective coordination mechanism is established. Coordination mechanisms in GSD are generally constituted by digital artifacts so that they are easy to share with remote team members. The establishment, development, and maintaining of social protocols is generally more challenging in GSD due to the lack of face-to-face communication. In the following, two examples are presented; the first describes the incident coordination mechanism used in the DHI case, while the second explains the establishment of the agenda coordination mechanism in one of the student projects.

Incident Coordination Mechanism. In the DHI team, one main coordination mechanism has been identified [22]: the tracking of development activities through Spira, a Test Management tool used as an issue tracker system. All development activities are tracked through Spira: the description of features to develop, the reporting of incidents, the assignment of tasks, and the description of test cases. Spira automatically assigns an "incident number" to all defects, test cases, and requirements that are the actual *artifacts* of the coordination mechanism. In the "Project handbook" document (part of the documentation available to the team) there is a diagram reported in Figure 1 that formally describes the workflow of an incident, explicating the *social protocols* that team members need to share in order to allow the coordination mechanism to function. The process is clearly defined: a defect that has been registered in Spira has to be approved by the PAM, who can reject it or evaluate it. During the morning meeting, Danish PAMs, acting as a Change Control Board (CCB), approved, planned, and assigned the incident to a developer through Spira. The developer that starts working on the defect changes the status to "in progress." and when he finishes working on that, he marks it as "completed." The developer then assigns the incident to the PAM, who tests and verifies it. The process defined in the diagram is supported by Spira, and the current status of every defect is visible in the tool at any time. Interestingly, the team members did not use the notification mechanism built into the tool, but notified each other using the Skype chat. During the feedback meeting, team members justified the seemingly more cumbersome social protocol with the opportunity for short social interchanges along with the notifications.

[10] The article was published before issue tracker were wide spread in industry.

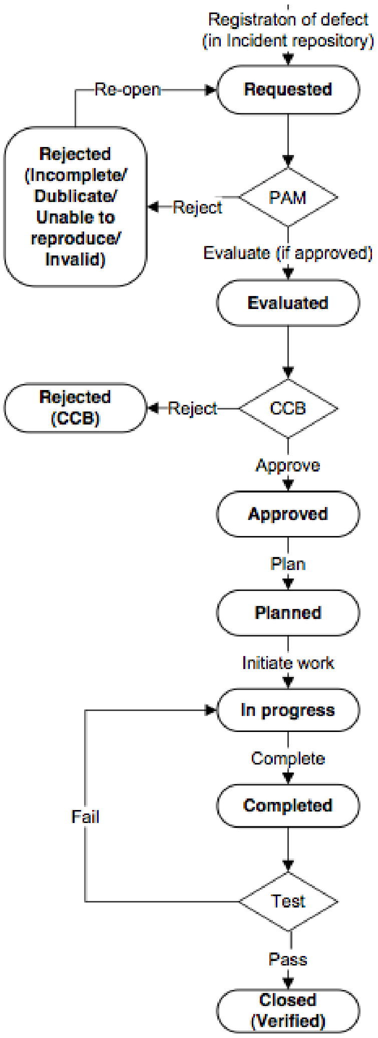


Figure 1: Incident Workflow. The diagram is part of the documentation used in the team.

Agenda Coordination Mechanism. In Team C [23] of the student team case, the Danish team members started to share an agenda for the meetings from the very beginning of the collaboration; the agenda was shared during the meeting through the Group Chat. After some weeks, a problem was reported in a Forum message by the Chinese team members in reaction to a not successful meeting: they explicitly requested to share the agenda prior to a meeting:

> We, PKU, advise that you, ITU, tell us agenda in advance. In that case, we, PKU, have time to prepare for it. And our meeting will be better. O(∩_∩)O *[Wusheng, China]*

In the same week, the Danish team members produced a document about the collaboration in which rules for the sharing of the agenda were proposed:

> A written meeting agenda should be uploaded at Wednesdays in order to accomplish effective virtual meetings. The author of these should be shifting between China and Denmark. So e.g. 1: one sub-team post a agenda proposal on Wednesday 2: the other sub-team confirm, write a respond to it or add things to the agenda latest 2 hours before the meeting on Friday. 3: the sub-team who originally created the draft will add a final agenda latest 1 hour before the meeting starts.

The suggested rules (i.e. proposals for social protocols) were discussed in a Forum thread and the coordination mechanism was established: after the agreement was reached, the agenda (*artifact*) was shared by one member at one location through a Forum message, it was integrated by a member at the other location, and finally modified by the members that proposed it (*social protocol*). This practice was carried out one after the other in the two locations during the remainder of the collaboration.

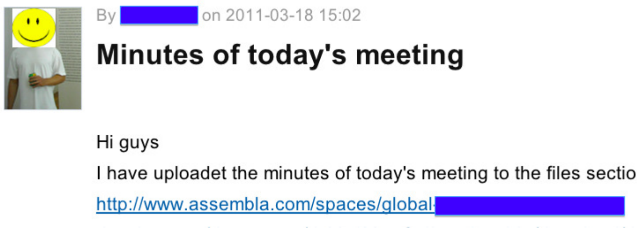
In summary. This section reported two examples of coordination mechanisms: (1) the *incident mechanism*, an established coordination mechanism formally defined in the documentation of an established team, and (2) the *agenda mechanism*, a novel coordination mechanism that was established in a forming globally-distributed team. In the former example, a collaborative SE tool - the issue tracker system - supports the coordination mechanism, while in the latter, the coordination mechanism is supported by communication through SoSo tools, namely IM group chat and Forum. In order to thoroughly describe how the communication occurring in distributed teams supports coordination mechanisms, the analytic concept of communicative genre is introduced and described in the following section.

5.2. Communicative Genres

5.2.1. Definition

Genre Theory states that genres create order to simplify the mass of available information [42]. Yates and Orlikowski have widely investigated the notion of *communicative genres* as a way for structuring practices in organizations [24]. A genre of organizational communication is a "distinctive type of communicative action characterized by a socially recognized communicative purpose and a common form" [43]. The communicative *purpose* of a genre is not rooted in a single individual's motive for communicating, but in a purpose that is constructed, recognized and reinforced within a community [44]. The *form* of a genre refers to the readily observable features of the communication, including structural features, communication medium and language [24]. *Structural features* can be text formatting, such as lists and headings, as well as devices for structuring interactions at meeting, such as agenda and chairperson; the *communication medium* can be pen and paper, face to face, telephone or mail; the *language* can be the level of formality or the specialized vocabulary used [24].

People produce, reproduce and change genres through a continuous process of negotiation and re-adaptation [43]. Genres may be considered at different levels of abstraction and can be combined in *genre repertoire*, a set of genres routinely enacted



By [redacted] on 2011-03-18 15:02

Minutes of today's meeting

Hi guys

I have uploadet the minutes of today's meeting to the files section:

http://www.assembla.com/spaces/global-[redacted]

Figure 2: Minutes Notifier Genre

by a particular community. A community's genre repertoire indicates "its established communicative practices and it can serve as a analytic tool for investigating the establishment of a community's communicative practices" [24]. Im et al. [45], for example, use genre and genre repertoire for analyzing email communication of a geographically-dispersed software team. In their work, they analyze and describe in detail the genres identified in the electronic communication between team members. It appears promising to adapt a similar approach and to perform communicative genre analysis also on other communicative channels used by distributed teams, such as SoSo.

5.2.2. *Example*

Many different communicative genres can be identified in our empirical cases. Genre analysis can be performed on mail, SoSo, but also on meetings and in general on any "type of communicative action characterized by a socially recognized communicative purpose and a common form" [43]. Our interest is in computer-mediated traces of communication, particularly in classifying the usage of SoSo in order to understand its role in GSD practices. An example of communicative genre recurrent in two different projects of the student case is reported in the following.

Minutes Notifier. In the student case, the Minutes Notifier genre is recurrent in Team A and in Team C, an example of Minutes Notifier genre is reported in Figure 2. The communicative *purpose* of the genre is to inform team members that minutes of the weekly meeting have been uploaded in the file section of Assembla. The *form* of the genre is similar in all Minutes Notifier messages identified:

- the *communication medium* is the Forum available in Assembla;
- the *structural features* make the genre easy recognizable: in the title the purpose of the message is reported, e.g.,"minutes of today's meeting" and it is followed by a brief message as well as by a link to the file uploaded;
- the *language* is usually based on specific terms, such as "minutes" and "file section" and goes straight to the point, with a brief informal opening like "hi guys."

The Minutes Notifier genre has been developed within two student teams (Team A and Team C) with the purpose of making all team members aware that a specific file, i.e., minutes of the meeting has been uploaded. An automatic functionality of Assembla enables a similar communication purpose, e.g., team members can receive an automatic mail whenever a file is uploaded. However, this practice has been established and, despite the fact that the coordination mechanism of file sharing is well supported by the file sharing system available in Assembla, an independent communicative genre is used to enact and reinforce the coordination mechanism. This appears to be a common relationship between communicative genres and coordination mechanisms; it will be described in detail in Section 6.

Please note that the purpose of the Minutes Notifier communicative genre is to inform team members that minutes of the weekly meeting have been uploaded, thus providing awareness to other team members. As we will show in Section 5.4, the awareness purpose underpins the situated articulation genre, which is a genre category that also includes the Minutes Notifier genre. Thus, a genre can be part of a genre category, sharing the purpose but differing in the form. The notion of purpose is further detailed in the next subsection.

5.3. *Communicative Purposes*

5.3.1. *Extending the definition*

In their analysis of communicative genres in mail, Orlikowski and Yates [24] identify purposes such as: response, question, proposal, for your information (FYI), and meta-comment. These purposes are not related to the specific practice observed and are not sufficient for the detailed analysis of concrete interaction through SoSo that we aim to perform. In our framework, the concept of *purpose* of a communicative genre is considered with a meaning related to the practice it is part of, rather than the one presented by Orlikowski and Yates in their studies, and the communicative genre analysis is performed in combination with interaction analysis [40]. Since, as mentioned above, communication genres can be identified at different levels, we consider purposes related to the practice context in which they occur, such as awareness and team building in order to understand the roles of communicative genres within collaborative software development practices. The response/question dimension used by Orlikowski and Yates [24] is maintained as a generic categorization that, when appropriate, specifies sub-genres with more specific purposes within the content related purposes identified.

5.3.2. *Examples*

In the student project reported in [23], several communicative genres, respectively, genre categories, are identified: work discussions, knowledge sharing, encouraging chats, social chats, metawork, and situated articulation. A detailed description of each communicative genre is outside the scope of this paper; however, few examples are reported in the following to clarify the perspective adopted in our framework and how it differs from the Orlikowski and Yates [24] approach. In particular, in the following, examples of *Work discussions* genre and of *Team building* genre are reported, both from the student projects case and from the DHI case. *Metawork* and *situated articulation* genres deserve a detailed discussion, as they are related with the articulation work concept introduced in Section 5.1. This is detailed in the subsequent subsection.

Work Discussions. Work discussions have the purpose of collaborating with remote team members. Work discussions can be related to decision-making, requirement specification, project planning, technical issues, etc. In our empirical cases, work discussions can be found in IM chats, Forums, and mails, with slightly different structural features and level of formality of the language due to the medium used, but with the same work purpose, and thus they fall in the same genre category. Work discussions usually start with a question, followed by one or more answers that sometimes lead to a broader discussion. The question/answer purpose suggested by Orlikowski and Yates [24] iis present as subcategory to add details to the purpose of performing the actual work, discussions about solutions to implement, or issues encountered. An example of a work question in the Forum of one of the student projects is reported below:

> CinemaService URL??
> Hey Rafael! What's the URL I should use to initiate the cinemaService?? *[Tommy, Team A]*

In the example, the work question is reported in the title of the forum message and in the post itself. The question is about a concrete work issue. One team member, Rafael, is addressed, despite it being a Forum message broadcast to all team members. In this way, if someone else in the team knows the answer, he can reply to the question; moreover, being in a Forum thread, the information is stored in a more persistent repository than, e.g., in an IM chat.

In the DHI team, work discussions while analyzing IM chat logs have also been highlighted in [22] and have been classified as collaboration chats. In the re-analysis through the communicative genre concept performed in this paper, the chats in [22] categorized under the purpose of collaboration fall in the genre category of work discussions. An example of an IM chat of the DHI case is reported in the following.

> *Jakob:* Does your solution for 699 hide any field that is of type GUID? The main reason this came up is because when you do an Intersect or Union, the GUID from the two input feature classes are added to the output table. This is what Joe wanted hidden, not just the standard "ID" field.
> *Naveen:* oh okay, I didnt do that :(
> *Jakob:* Yes, I think the criteria should be if the field type is "Guid", not what the field name is.
> *Naveen:* but cant some other guid fields could be important.
> *Jakob:* Possibly, but for now we want them all hidden.
> *Naveen:* okay will do that.

In this example, it is possible to see how work discussion chats complement formal design specifications. Team members refer to the incident number (699) and discuss the solution of an incident, clarifying what the software should do under specific conditions. The work discussion starts with a question and the technical issue is solved with very fast interactions between co-workers, probably faster than detailing it in formal design documents. Decisions that occurred in the chat but should be reported in the issue tracker system in order to document them are included in the issue tracker after the chat. From the log analysis, it is visible that when the topic is too complex for a written chat, the conversation that started in IM is then usually moved to a call. Moreover, if issues are considered to have a wider implication, team members may start a discussion on e-mail [22]. These findings have been confirmed by the Danish team members during a workshop, indicating that well-established teams have different genres for the same overall purpose, depending on, for example, the complexity of the topic and the level of impact of the discussion at hand.

Team Building. In the analysis of the student project communicative genres [23], we identified two different genres with the purpose of team building: socialization genre and encouraging chats. These genres cannot be related to any question/answer purpose proposed by Orlikowski and Yates [24]. However, they appear fundamental in communication between team members, both as an explicit purpose and as a sub-text [22] in other genres, e.g., work discussions.

Socializing chats, referred to also as cheap talk [46], consist of messages with the sole purpose of socializing with the remote team members talking, for example, about each other's country; an example of IM chat occurring in the Team C of the student project is reported below:

> *Zhang:* Denmark is a beautiful country. I will go to Denmark to have a tour if there any chance, To got to know this country
> *Stella:* well please come... It is beautiful!! I have several chinese colleagues and friends at work (sushi restaurant)

This chat occurred in the beginning of the collaboration, when IM chats appear to be used to get to know each other and to establish a good working environment, as confirmed during the post-mortem interview with Stella.

In the DHI team, pure socialization chats are rare and are mostly used as introductory socializing talk at the beginning of the day, as it is reported in the following example.

> *Prashant:* Hi Jonas
> *Jonas:* hi Prashant
> *Prashant:* hows monday treating you?
> *Jonas:* better than usual. Working from home
> *Prashant:* oh we r on the same boat
> *Jonas:* good to hear :)

The socialization chat provides awareness about the situation of the remote colleague Jonas, with whom Prashant works daily;

the chat appears as an opening to establish the connection for further communication during the day.

Encouraging messages are positive feedback to a message posted by another team member, which can belong to any other genre. Two examples from the student projects are provided below:

> You did a great job! 佩服你们。 And we will finish our part of self-introduction soon. *[Yang, Team C]*
>
> This is really cool:] *[Arnold, Team B]*

The two examples clearly show appreciation for the work performed and encourage remote team members to continue the successful collaboration. In the first case, some Chinese characters are reported as a translation of the English sentence "You did a great job" — thus adding a funny and challenging connotation to the message. Encouraging chats in the student teams happen both in IM and in Forum.

In the DHI team, encouraging IM chats are very frequent. In the following example, after a technical discussion, the manager gives encouraging feedback to the developer and shows that he appreciates the work that has been done.

> *Julius:* uh - you are very methodic. That is great
> *Shahid:* Thanks
> *Shahid:* but if i have missed any..then please let me know
> *Julius:* I will - but please home now :)

The example shows Julius' appreciation for the work performed by Shahid and the commitment of Shahid. The tester is working until late in the evening to finish his job and he wants to be sure he has done all tests necessary, "If i have missed any..then please let me know." Julius assures Shahid and pushes him to "go home" acknowledging the commitment of the tester. These encouraging chats allow team members to maintain a good working relationship despite geographical distance. The importance of encouraging chats is highlighted by one of the PAMs of the DHI case:

> Our team is not so big, so I think it's important to take care about what is happening and give the right attention to each team member. *[Jakob, Denmark]*

Summarizing, socialization chats are visible, mainly through IM, in the initial phase of the student project and as an initiation of the working day in the industrial case. Encouraging chats are common, both in the student case and in the industrial case through IM or Forum messages, and are usually reactions to successful achievements of the remote team members. Depending on the maturity of the team, different kinds of team building conversations can happen and they seem to establish, support, and improve the collaboration across sites. The supportive role of team building chats for the development of other communicative genres and for the success of the collaboration is further discussed in Section 6.4.

5.4. Articulation Work

The articulation work concept has been introduced in Section 5.1, in relation with the definition of coordination mechanisms. However, articulation work can be understood as a communicative genre category. Thus, it deserves a particular attention, and an extension of the definition is provided in the following, supported by some examples.

5.4.1. Extending the definition

Schmidt and Simone [3] refer to the concept of *articulation work* that is defined as:

> "...a recursive phenomenon in that the management of an established arrangement of articulating a cooperative effort may itself be conducted as a cooperative effort which, may also need to be articulated." [3]

In cooperative work settings characterized by complex task interdependence, "coordination mechanisms reduce the complexity of articulation work and alleviate the need for ad-hoc deliberation and negotiation" [3]. In our framework, two levels of articulation work are distinguished, as suggested by Gerson [47] and Strauss [48]: metawork and situated articulation. *Metawork* is used to describe the development of a social protocol. *Situated articulation* denotes the discussion of the state of the current task in order to coordinate this task; situated articulation involves "adapting a social protocol to a situated use" [49]. Metawork and situated articulation can be understood as categories of communicative genres that are used for changing and articulating not only the coordination mechanisms but also the communicative genres of a project team.

5.4.2. Examples

Despite the fact that coordination mechanisms should alleviate articulation work, it becomes obvious that articulation work in the form of metawork is necessary to establish coordination mechanisms and that, once the coordination mechanism is established, further articulation work in the form of situated articulation appears fundamental to support the coordination mechanism. This section provides some examples to show the difference between the two kinds of articulation work, while Section 6 will detail the supportive role of metawork and situated articulation for establishing and enacting coordination mechanisms.

Metawork. Metawork communication consists of metacomments about how to structure the work within the team. This can be in the form of proposals, questions, answers or discussions about how to carry on the work. Metawork allows establishing coordination mechanisms. An example of a metawork proposal that occurred in the Forum of one of the student teams is reported below:

> Hello, everyone of ITU, I made a table by Word. In order to find the best time for our meeting, please download the .doc file and fill the table, and then upload it in a reply message. By it, we can have a clearly

> view which is the proper time. Thanks! *[Cheung, Team C]*

In the Forum metawork message, Cheung suggests a procedure to follow (*social protocol*) for using the word document (*artifact*) to find the best time for the weekly meeting. Thus, a temporary coordination mechanism is established by Cheung, who also suggests the social protocol to follow.

Metawork is supported both by IM chats and Forum messages, helping to establish persistent coordination mechanisms, as shown in detail in the temporal analysis performed for Team C and reported in [23]. Thanks to metawork, effective negotiation of social protocols and decisions about coordination mechanisms have been made between team members and metawork discussions usually end when coordination mechanisms get established.

In the DHI case, metawork does not appear in the IM conversations: team members just do things, without discussing how the work needs to be performed. It is possible, however, that metawork took place in channels other than IM in order to inform all team members of practices to follow, e.g., in mails. However, we did not analyze in detail the mail communication of the DHI team, thus only hypotheses can be suggested.

Situated Articulation. Another kind of articulation work, situated articulation, is visible in communicative practices of both the industrial and student case. Situated Articulation communicative genres are questions, answers or FYI messages about the work that has been done so far with the purpose to provide awareness to all team members about the status of the project. In particular, team members can ask for an update about the status of the project, they can reply or independently provide an update of what they did recently. The Situated Articulation genre is particularly interesting because it is used for supporting and enacting established coordination mechanisms.

In the three student projects, a large amount of the *File Notifier* genre is reported as an example of situated articulation genre category. The *File Notifier* genre is a broader genre that includes the *Minutes Notifier* genre described in Section 5.2.2. File Notifier messages consist in notifications about the sharing of files, minutes, agenda, and the standup meeting that support the associated coordination mechanisms. It is interesting to note that in Team C, once coordination mechanisms have been established, patterns of collaboration are clearly visible in the Forum, e.g., as sequences of messages: standup, agenda, and minutes [23].

In the DHI case, many situated articulation chats were observed. In [22], the authors referred to these chats as coordination chat; however, the coordination dimension is present in the coordination mechanism itself, while the situated articulation chat has the purpose to provide awareness, supporting the coordination mechanism. An example of situated articulation chat in which the social protocol is "adapted to a situated use" is reported below.

> *Prashant:* Hi Jonas..
> *Prashant:* I have implemented . . .
> *Prashant:* the implementation is working fine
> *Prashant:* do u want me to check-in the code and test on ur system??
> *Jonas:* maybe you could show it to me ?
> *Jonas:* I'll call when Morten is ready
> *Prashant:* ok.call me once you guys are ready

In this example, Prashant reports that he has completed a task and he makes Jonas aware that the "implementation is working fine." The fact that the task is completed is reported in Spira, the issue tracker system, and, following the incident workflow (see Section 5.1) Jonas should check the implementation without the explicit request of Prashant. The coordination mechanism is established in the team and theoretically does not require further communication. However, since the incident is of particular relevance, Prashant wants to comment on the finalization of the task, and he wants to be sure that the implementation matches Jonas' expectations. Therefore, a situated articulation message is sent to Jonas to adapt the social protocol defined for the coordination mechanism to the specific case, highlighting the importance of the incident solved. A broader discussion on the establishment, adaptation and maintaining of social protocols is carried on in the following subsection.

5.5. Social Protocols

5.5.1. Extending the definition

The definition of coordination mechanisms (see Section 5.1) explicitly mentions the role of the social *protocol*: "a set of rules, conventions, policies shared by people involved in the cooperative activity" [3]. In the communicative genres definition, protocols are not defined; however, it is reported that *social norms* are visible in recurrent communicative situation [43]; thus, social norms are underpinning also the concept of communicative genres. More in general, the concept of *social protocol* (or social norm or social rule) is related to the concept of *practice* and, particularly to the social nature of it, as stated by Wittgenstein [50]:

> . . . *obeying a rule* is a practice. And to think one is obeying a rule is not to obey a rule. Hence it is not possible to obey a rule *privately*: otherwise thinking one was obeying a rule would be the same thing as obeying it. (P.I. 202 [50])

The social nature of the norm thus comprises the necessity to be shared by people involved. Wittgenstein also suggests that [50]:

> . . . we lay down rules, a technique, for a game, and that then when we follow the rules, things do not turn out as we had assumed (P.I. 125 [50])

That is to say, social protocols not only need to be initially *decided* upon, but also *adopted* and *adapted* by people over time, thus being socially *shared*, *modified* and *appropriated*. This phenomenon is also referred to as local and temporary alignment of social practices, see e.g., [51], [52]. As visible in our field material, social protocols come about and are maintained in different ways. Social protocols can be both explicitly and implicitly defined by team members. A social protocol is *explicitly* defined through metawork when team members discuss whether and how to use a specific artifact. For example, a team can decide to share an *agenda* (the artifact of the coordination mechanism) prior to a meeting and how team members in each locations are alternatively responsible for that. A social protocol can be *implicitly* adopted, e.g., based on previous *work experience* in the project or from *professional knowledge*. The social protocol on how to use the agenda during the meeting might be *implicitly* driven by *previous experience* of team members and does not require further discussions on how to use the artifact. A social protocol can also be *defined* but *not adopted*, when team members agree on it, but later do not use it. Social protocols are formed, negotiated, established and maintained by team members in globally-distributed teams, thanks to the usage of SoSo. Some examples from the DHI case are reported in the following.

5.5.2. Examples

In the DHI case, both Word and PDF documents are used to formalize the planning and the execution of the WB project. The Project Organization and Project Life Cycle are formally defined in a document with guidelines for the Software Development Project. The document incorporates a short, step-by-step guide on how Spira should be used by the team to manage software development projects. The social protocol on how to use Spira is formally described in the documentation, and is defined, shared, and established among team members. From the observations during the project, the researcher can confirm that the social protocol was also adopted by team members. Despite the social protocol being well established, some IM chats present in the field material show how social protocols need to be maintained between team members. An example of situated articulation chat is reported in the following.

> *Naveen:* any incident you have planned for me...
> *Naveen:* to fix
> *Jonas:* look in SPIRA for planned - take from the top with respect to priority

Even though the social protocol is known by Naveen, he explicitly asks Jonas if there is something in particular he should work on, wanting to double check whether or not to rely on Spira — thus on the coordination mechanism. Jonas answers, confirming following the social protocol through which this chat is reinforced and maintained. The need for confirmation of an already defined social protocol from the Indian developers is an aspect that emerged during the workshop with Danish team members: these kinds of chats serve to maintain a social relationship with remote team members, thus by PAMs they are considered to be important, though disruptive.

In the documentation, the role played by the "incident number" assigned by the tool to any issue inserted in the system is formally described. However, the number is also used in the check-in comments of the source code, as defined in a less formal way in a piece of paper hanging on the wall of the Indian site that summarizes the "todo before checking in the code." In this case, the social protocol on how to use the issue number in the check-in comments is defined, shared, established, and adopted by team members.

Another social protocol underpins the use of the incident numbers. Team members continuously refer in the IM chats to the incident number, as shown in the following example.

> [10 : 07] *Arun:* 974 is yours now for verification (gap filler scrollbars)
> [10 : 09] *Nelson:* thanks
> [11 : 58] *Arun:* 1008
> [12 : 07] *Nelson:* all yours
> [12 : 57] *Arun:* your now.
> [13 : 18] *Nelson:* 847 is yours

In this case, a situated articulation chat is carried on between the two team members that exchange incident numbers in the chat, along with assigning the task in Spira, and the incident numbers bouncing back and forth, allowing fast coordination between team members. In the example, time stamps are reported, indicating that collaboration took place during the whole morning; team members rely on each other to know what to do, and the shared social protocol helps them understand what to do when they receive an IM chat with a number. The situated articulation chat augments the coordination mechanism enacted in Spira, while the social protocol is undefined, implicitly adopted based on previous work experience, and then shared, which allows team members to understand the meaning of the number mentioned in the chat.

5.6. The Repertoire

In this subsection, the concept of repertoire, defined for communicative genres, is described and adapted to the set of coordination mechanisms used in the teams.

5.6.1. Extending the definition

Genres may be considered at different levels of abstraction and they can be combined in *genre repertoire*: "a set of genres routinely enacted by a particular community" [24]. A community's genre repertoire indicates "its established communicative practices and it can serve as a analytic tool for investigating the establishment of a community's communicative practices" [24]. The concept of repertoire can also be used in relation to the coordination mechanisms, as teams can establish a repertoire of coordination mechanisms that indicate the coordinative practices enacted in the team. In [22], the authors refer to the concept of ecology of channels, comprising both the concepts of repertoire of communicative genres and the concepts of repertoire of coordination mechanisms. However, a distinction between the two is useful for understanding how the repertoire develops during the project, how it gets established, and what the relations are between the two in a project. Some examples are reported in the following.

5.6.2. *Examples*

The student projects were examined by the researchers during the whole collaboration and team members gave full access to all artifacts produced and tools used; thus, it is possible to provide an overview of the repertoires developed in the teams during the whole collaboration. The first example refers to a previous paper [23] describing the repertoire of coordination mechanisms developed in Team C of the student case. The second example is provided to show that it is possible to compare different repertoires: in particular, a comparison of the communicative genres repertoire in the Forum of the student projects is illustrated. The third example describes the role of IM as a dispatcher in the repertoire used by the DHI team, as team members are aware of the repertoire of tools they use, and they share social protocols on how to use them; this example refers to previous work [22] reported in this paper which introduces the relation between the communicative and the coordinative repertoire.

Repertoire of Coordination Mechanisms. In [23], the authors describe in detail the establishment and development of the repertoire of communicative genres in Team C of the student project case and they relate it to the establishment of the repertoire of coordination mechanisms. Five coordination mechanisms are established in the team: the issue management mechanisms, the versioning of the source code, the file sharing mechanism, the standup meeting, and the sharing of the agenda for the meetings. All coordination mechanisms are supported by functionalities available in Assembla. However, while the first three mechanisms are supported by tools traditionally used in Software Engineering, such as the issue tracker tool, the version control system, and the file sharing system; the last two coordination mechanisms, the standup meeting and the agenda, are supported by the Forum. A detailed description of all coordination mechanisms is provided in the paper, as well as the supportive role of the communicative genre repertoire for the negotiation and establishment of the repertoire of coordination mechanisms [23]. In summary, the empirical results of the paper show that the role of SoSo is to support informal communication, enabling social talk and metawork, necessary for both establishing and maintaining effective coordination mechanisms, thus successful cooperation [23].

Repertoire of Communicative Genres. A detailed analysis of the whole communication through Forum messages in the three student projects has been performed through the communicative genre analytic concept. This is to provide a quantitative overview of how the communication occurred within the three distributed teams, thus allowing the comparison of the different repertoires developed. Five communicative genre categories have been identified: team building, situated articulation, metawork, work, and knowledge sharing. A description of each genre category is provided in [23]; most of the genres have also been explained in the previous subsections. Once SoSo logs have been coded through genre analysis, the numerical distribution of genres can provide some insights about the teams, as reported in Figure 3, indicating which communicative genres are developed more often by each of the three student teams. Please note that in our field material, one Forum message is coded with one single genre. If more genres appear to be suitable for a message, the main genre is assigned. Figure 3 gives some indications about the amount of communication that occurred in the different teams, particularly highlighting that Team C invested a lot in team building and articulation work discussions, while in Team A metawork and team building genres did not occur as often. However, analyzing only the repertoire of communicative genres does not provide insights into the collaboration. By relating the communication repertoire with the repertoire of coordination mechanisms, it is possible not only to provide a description of practices, as the authors have done in [23] but also to show the hypotheses on the possible reasons of breakdowns, as discussed in Section 6.4.

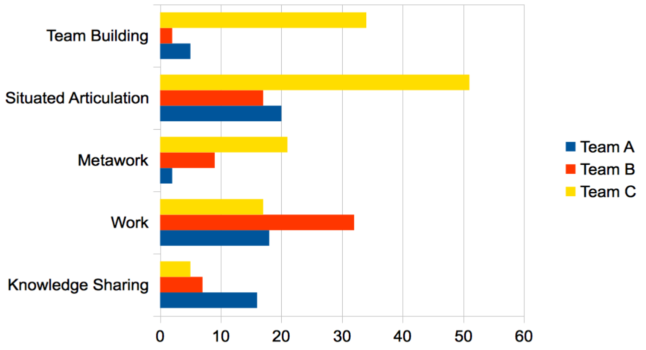


Figure 3: Repertoire of Communicative Genres in the student case. The graph shows the number of instances for each communicative genre identified in the Forum communication between team members of the three teams under study.

The dispatcher role of IM chats. In the DHI case, a full analysis of the repertoire of coordination mechanisms and of communicative genres has not been performed. Some coordination mechanisms have been identified in [22] and have been reported in this paper to show how coordination mechanisms work effectively in an established team. A quantitative analysis of the communicative genre repertoire has not been performed, as no reference point was available to interpret the results. However, from the sample of IM chats coded for the purpose of this paper, it is clear that IM chats belong mostly to the situated articulation genre and to the work genre; metawork discussions rarely occur in IM chats, and encouraging chats are more recurrent than are pure socializing chats. It would be problematic to conclude anything based on the different distribution of communicative genres alone: the DHI team did not use any team chat. For metawork, which necessarily includes the whole team, other channels, especially e-mail, seems to have been used. E-mail communication was not in the focus of that fieldwork and was therefore not collected. We can, however, affirm the wide usage of situated articulation messages indicating that coordination mechanisms are effectively established in the team. Team members use the repertoire and are aware of how to use the different tools, and thus they share common social protocols. Our analysis supports the fact that in distributed settings it is important not only to use the same set of tools, but also to develop common social protocols through articulation work and metawork. In particular, in [22] the authors refer to the concept of "ecology of channels," to indicate a socio-technical communication system where different channels are used in a complementary way [22]. The analysis provides an indication that IM plays a special role in such a socio-technical system, acting as glue between different communicative channels and complementing collaborative SE tools that provide templates for coordination mechanisms. These aspects will be further discussed in Section 8.3. In the following, the relation between communicative genres and coordination mechanisms is described in greater detail.

6. Relation between Communicative Genres and Coordination Mechanisms

The mutually supportive relation between communicative genres and coordination mechanisms is evident in many of the examples reported in Section 5, and it has already been highlighted in several cases. However, the relation warrants a more detailed analysis. In this section, more complex examples are analyzed in order to motivate and explicate how the theoretical framework presented in Section 7 comes about.

6.1. Example 1: File Notifier genre and File Sharing mechanism

In the coordination mechanism repertoire of the student projects, described in Section 5.6, the *file sharing coordination mechanism* is reported as part of the coordinative repertoire established in Team C. In the file sharing system available in Assembla, important files (*artifacts*) are regularly exchanged, and, when the file is considered to be of particular relevance, a message is posted in the Forum to notify other team members that the file is available in the system (*social protocol*). The Forum message constitutes a *File Notifier genre* described in Section 5.4, as part of situated articulation genre category occurring in the student teams, and, in this case, supports the coordination mechanism through explicit communication mediated by SoSo. Whenever coordination mechanisms are established, the situated articulation genre in SoSo sustains the coordination mechanism, allowing team members to keep an informal channel open for further discussions or for social talk. Another example of this kind of situated articulation genre that enacts and enhances the coordination mechanism is visible in the DHI case and is reported in Section 5.5: situated articulation chats containing incident numbers are exchanged between team members, sustaining the established coordination mechanism of issue tracking supported by Spira.

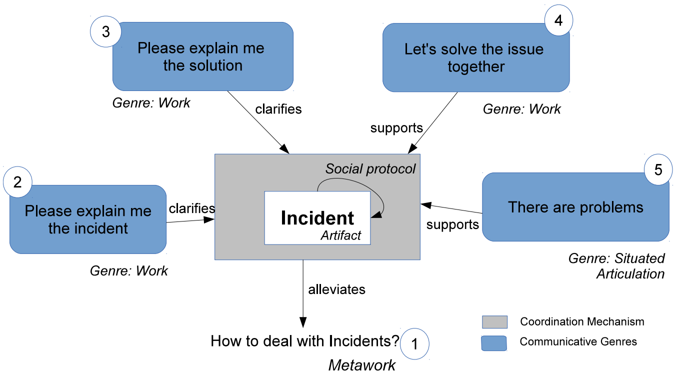


Figure 4: The Incident Coordination Mechanism is supported by Work Genre and Situated Articulation Genre.

6.2. Example 2: DHI incident

In Section 5.1, the *Incident Coordination Mechanism* established in the DHI team was described. The process described in the incident workflow available in the documentation is supported by communication through mails, voice calls, screen sharing, and IM chats. Informal communication takes place in order to clarify the formal specification, adding information to the descriptions reported in Spira: different communicative genres support the Incident coordination mechanism. Following the lifecycle of a specific incident, we identified three phases in which communication took place across sites [22] when the defect was (a) assigned to the PAM, (b) when the developer worked on it, and (c) when it is in status "completed" (see Figure 1 for the representation of the incident workflow).

Figure 4 shows a representation of the *Incident coordination mechanism* and of the communicative genres that support it. The coordination mechanism alleviates the necessity of ad-hoc discussions on how to deal with incidents (1). In phase (a), if a tester or a developer assigns the defect to a PAM, the steps to reproduce the incident described in Spira could be insufficient for the PAM to reproduce it. In this case, the PAM starts an IM chat with the Indian member through IM (*work genre*), or he may ask to start a screen sharing session in order to better understand the problem (2). When a developer is working on the resolution of an incident (phase b), and he has doubts on how to implement the solution described in Spira, he contacts the PAM to discuss the technical issues (*work genre*), through IM or audio (3). Finally, when an incident is closed (phase c), it can happen that the implemented solution does not completely solve the issue, or the solution can interfere with other parts of the software. In the former case, the PAM uses chats and audio to talk directly with the developer to solve the issue (*work genre*) (4) or he may decide to re-open the incident (*situated articulation genre*) (5). In the latter case, a mail is sent to all PAMs and developers involved to inform all members about the problem and to find a shared solution (*situated articulation genre*) (5). The resolution of incidents takes place not only through the coordination mechanism enacted by Spira and by the source code repository, but it also involves a whole range of communicative genres supported by different communication channels: mail, chat, and screen sharing that sustain the coordination mecha-

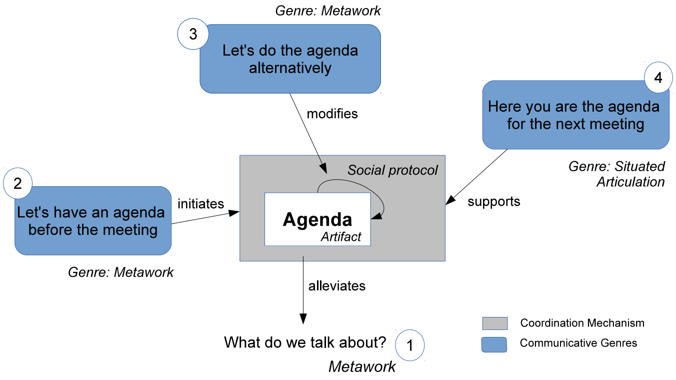


Figure 5: The Agenda Coordination Mechanism is supported by Metawork Genre and Situated Articulation Genre.

nism, thus allowing the success of the collaboration. Among the different channels used, IM chats comprise the initial dispatcher through which other channels are then considered and used, if needed.

6.3. *Example 3: student agenda*

Section 5.1 has described how the *agenda coordination mechanism* is established in Team C of the student case. Figure 5 shows a more complete representation of how the coordination mechanism is established, highlighting the relationship between communicative genres and coordination mechanisms. The agenda is a coordination mechanism that alleviates the articulation work needed to decide what should be discussed during a meeting (1). In reaction to an unsuccessful meeting, the team discusses the usage of agenda in a Forum thread and decides to share the agenda before the meeting, thus the *metawork genre* serves to initiate the coordination mechanism (2). Once the coordination mechanism is defined and adopted, a re-discussion of the protocol occurs through Forum messages, indicating that the agenda should be produced by team members in both locations, alternating each week; thus, the social protocol evolves over time, thanks to *metawork* (3). Once team members have decided who is responsible to share the agenda before the meeting, the responsible person creates a Forum message that is broadcast to all team members communicating the agenda of the week; thus, *situated articulation genre* takes place in the Forum as part of employing the coordination mechanism, supporting and enhancing it (4). The coordination mechanism is initiated, modified, and enacted through two different communicative genres: metawork and situated articulation, both supported by Forum messages.

6.4. *Example 4: comparing three student projects*

Section 5.6 has reported the analysis of the repertoire of communicative genres appearing in the Forum messages of the student case, allowing comparison of the three teams. However, it is necessary to relate the repertoire of communicative genres with the repertoire of coordination mechanisms established in each team in order to describe the different teams and to show possible reasons for breakdowns. A detailed analysis of the communicative and cooperative practices in the three student teams is reported in the following and a synthesis is provided at the end of this subsection to provide insights on the collaboration.

Team A. Figure 3 shows that team members scarcely communicate through the Forum, mostly using it for situated articulation and rarely for metawork or team building. Most of the conversations were started by the Danish side; seldom did Brazilian students use the Forum, mails or IM chats to communicate with the remote team members. The team mostly relied on weekly meetings and on some coordination mechanisms. In particular, the team effectively established one coordination mechanism: the sharing of the agenda supported every week by a situated articulation message that was posted in the Forum alternatively by team members in each location. Many of the other coordination mechanisms did not seem to be used in a common way across sites. For example, the file sharing coordination mechanism was used by team members in both locations, but the social protocol was not shared between team members: the File Notifier genre through Forum messages was used solely by the Danish team members. An example of coordination mechanism that was initiated, but not adopted, is the issue managing system that was initiated by the Danish team members, creating 50 issues throughout the project; however, the system was never really used by the Danish members, nor by the Brazilian members. The lack of communication and of shared social protocols caused misunderstandings and major challenges in the last part of the project, when the two parts of the code — developed independently by the two sub-teams — needed to be integrated. An example of issues related to the lack of articulation work is reported in a metawork Forum message:

> If anybody else engage in testing of the remaining components, then it would be nice if you would post a message so we don't do double work and create conflicts in SVN as Roberto and I did earlier today :-) *[Jonas, Denmark]*

The collaboration of Team A is proven to be unsuccessful, as is evident in the field material and reported by team members in the final report; many reasons could be given to explain this. The analysis of communicative and coordinative practices can provide some indications about the misalignment between communicative genres and coordination mechanisms. Metawork did not take place in any tracked communication tool, as it occurred only during weekly meetings rather than in mail or SoSo, and coordination mechanisms did not effectively get established between team members. Metawork certainly occurred during meetings; however, it did not appear sufficient to get a shared understanding of the negotiated practices to follow between team members. Moreover, the team did not invest in team building through SoSo. Other teams, e.g., DHI case and Team C, seemed to encourage and sustain the collaboration. To conclude, from the analysis of Team A, we can affirm that relying only on pre-established coordination mechanisms does not appear advisable, especially if social protocols on the usage of the coordination mechanisms are not successfully negotiated and shared between team members.

Team B. The collaboration started with many challenges and team members tried to overcome them by imposing processes and social protocols to improve the collaboration through many metawork proposals (see Figure 3). However, negotiation and shared adoption of practices did not occur, and imposed processes were not followed by all team members. For example, daily stand up meetings were proposed to give the project an "everyday" effect; however, they were used rarely, since, as stated in the final report of the Danish sub-team: "they appeared to be a quite formal requirement." Team members did not invest in team building, and the metawork was not intertwined with the actual work, as is evident in Team C; this can be seen as one of the factors influencing the unsuccessful establishment of common practices. As the project progressed different social protocols were established from what was explicitly decided in the beginning, and the initial metawork was replaced by practices derived from the actual work. Thus, some explicit coordination mechanisms were established in the last part of the project; for example, the issues managing system was used only in the last month of the collaboration (22 issues in total), despite software development and collaboration activities already taking place in the previous months. The team often used the File Notifier genre reported as part of situated articulation in Figure 3 to highlight the uploading of some important files; however, no regular patterns were visible, as reported for Team C. As Figure 3 shows, most of the messages exchanged in Team B were work discussions. Since both sub-teams had responsibilities for all parts of the software, many dependencies in the actual collaboration arose and team members needed many work discussions. Especially in the last part of the project, the Forum was effectively used to support them, and its asynchronous nature helped to overcome the lack of overlapping working hours. However, the huge number of work discussions could also be an indication of the scarce effectiveness of the coordination mechanisms adopted. In summary, similar to Team A, Team B metawork did not turn out to be successful; certainly, the team did not use SoSo to push team building conversations to establish a good environment for the collaboration. Nevertheless, the team finally succeeded in the collaboration, despite encountering difficulties in the integration of the software developed by the sub-teams bringing about the required intense collaboration in the last part of the project.

Team C. Figure 3 shows that much more communication through Forum is visible in Team C than in other teams. In particular, the team had many team building, metawork, and situated articulation chats, all of which had a positive influence on the actual collaborative work performed. The work discussions were mainly performed during weekly meetings in the weekly standup, as well as in the group chat. In the team, socialization and encouraging chats were mixed with the actual work. Team members made effective usage of many coordination mechanisms: 110 issues were produced and used as the main coordination mechanism for the software development activities. The large number of File Notifier messages, reported as part of situated articulation in Figure 3, demonstrates that the sharing of minutes, the agenda, and the standup meeting worked as effective coordination mechanisms, which were adequately supported by the situated articulation genre. For example, the Minutes Notifier genre supported the coordination mechanism of sharing the minutes of the meetings. Producing and sharing the minutes of the meeting was an initiative of the Danish team members and was not a practice commonly defined or requested; however, it became an established practice. Chinese members considered it very important, since it helped them "to go through the discussions that occurred in the meeting and check if there was a common understanding of what was discussed." It is interesting to note that, in Team C, practices were established after the first month of collaboration, from which patterns of collaboration were clearly visible in the Forum, e.g., as sequences of messages: standup, agenda, and minutes. Most of the situated articulation messages effectively supported the different coordination mechanisms established. The development of the repertoires in Team C is described in detail in [23]; however, from the description provided, there is evidence of the supportive role of the Forum for initiating, re-negotiating (through metawork), and sustaining (through situated articulation) the coordination mechanisms. The high amount of team building seems to have an impact on the cohesion of the team that supported the collaboration.

In summary. The communicative and coordinative practices of the three student projects have been described in detail in the previous paragraphs using the concepts of communicative genres and coordination mechanisms in a complementary way. Thanks to the analysis performed, it is possible to hypothesize possible reasons for the success of the teams or for the breakdowns.

Team A relied solely on professional coordination mechanisms, without dedicating time and effort on metawork for the negotiation and establishment of shared social protocols, nor spending time on team building for encouraging a good work environment; thus, the team experienced major challenges in the final phase of the project. In Team B the initial metawork also did not turn out to be successful, and no team building occurred. The team did finally succeed in the collaboration, though encountering difficulties in the integration of the software developed in the last part of the project. Team C succeeded with the establishment of shared coordination mechanisms and effective social protocols, thanks to the initial successful metawork and to team building. In Team C, metawork and situated articulation were fundamental to establishing, maintaining, and enacting coordination mechanisms that allowed the establishing of smooth collaboration and a positive working environment.

The hypothesis is thus that the development of satisfying communication and coordination practices depends on the successful establishing and maintaining of social protocols, which seems to be dependent on the initial metawork and on team

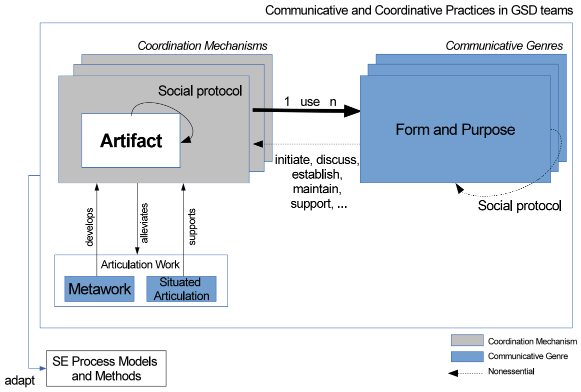


Figure 6: The Conceptual Framework: Coordination Mechanisms and Communicative Genres can be used to describe Coordinative and Communicative practices in globally-distributed teams. A Coordination Mechanism uses one or more Communicative Genre and Communicative Genres can support the establishment or the maintenance of a Coordination Mechanism, e.g., by reinforcing it. Metawork and Situated Articulation are special Communicative Genres, which negotiate, develop, establish and support coordination mechanisms. In this way, Articulation Work is alleviated by effective Coordination Mechanisms.

building. However, these hypotheses that arise from the analysis performed need to be confirmed through further investigation. These aspects will be further discussed in the next section.

7. The Conceptual Framework

This section describes the conceptual framework that we propose for analyzing and describing coordinative and communicative practices in GSD, relating the notions of coordination mechanisms [3] and communicative genres [24] described and extended in Section 5. For the authors of the present paper, a framework is a set of related concepts that is used to analyze cases and to explain phenomena. Relating concepts together in a single framework, it is possible to describe not only one specific case but several cases of a different nature. Thus, a framework allows generalizing from empirical cases to general phenomena.

The framework proposed, reported in Figure 6, shows that existing software process models and methods need to be adapted and appropriated by team members in situated action [13], as is also suggested in the framework proposed by McChesney and Gallagher [39]. However, our framework is based on the fact that the development of common social protocols is crucial for achieving effective communicative and coordinative practices, as well as for the adaptation of models and methods. The concepts of coordination mechanisms and communicative genres are both based on the notion of social protocols and are mutually supportive; thus, they allow to describe how communicative and coordinative practices are developed and maintained in distributed teams. A coordination mechanism consists of a coordinative protocol imprinted upon a distinct artifact [3], while a communicative genre constituted by a form and a purpose can be established, evolving over time, thanks to social protocols shared between team members.

In summary, coordination mechanisms provide a way to collaborate through digital artifacts, alleviating metawork. However, they are supported by explicit communication, thus by communicative genres. Thanks to communicative genres, coordination mechanisms can be initiated, discussed, established, and maintained, thus able to support the collaboration between remote team members. Section 6 provides several examples that describe communicative and coordinative practices in different globally-distributed teams, both novel and established ones, using the theoretical concepts of this framework and showing the mutually supportive relationship between communicative genres and coordination mechanisms.

The novelty of the framework proposed, compared with McChesney and Gallagher's [39], is the usage of coordination mechanisms and communicative genres to illustrate the adaptation of models and methods in practice. Moreover, the framework is developed while analyzing the field material from two empirical cases, showing how the actual communicative and coordinative practices can be described by using the two concepts and highlighting the supportive role that they play for each other. Finally, the framework shows the development of practices, highlighting how specific communicative genres come about, thus offering a way to describe the different genres, such as metawork or situated articulation, that occur in SoSo. The communicative genres identified are more specific than the one proposed by Yates and Orlokowski [24], as they are employed in combination with interaction analysis [40] and are used to investigate in detail the interaction occurring in the communication within distributed teams.

8. Discussions

This section presents and discusses the main contributions of this article. Initially, the usefulness of the conceptual framework is established and the importance of repertoires is highlighted, after which the role of SoSo in the repertoire is discussed. Finally, implications for research and for practice are reported, and limitations are presented, together with future research directions that arise from the findings of this paper.

8.1. On the Conceptual Framework

This paper presents a novel conceptual framework that integrates and extends the theoretical concepts of coordination mechanisms and communicative genres, showing how they mutually support each other. Coordination mechanisms alleviate articulation work; moreover, communicative genres with the purpose of articulation work can initiate, redefine, enhance, and support coordination mechanisms, alleviating the necessity of further and more complex articulation work. Using the framework, we can see in greater detail when and how successful metawork takes place, resulting in commonly adopted coordination mechanisms that lead to an understanding of the situations in which coordination mechanisms fail. The theoretical framework is constructed upon the empirical material of the cases described; however, it can also be used to describe practices in other GSD projects. For example, a study by Damian et al. [53] shows that coordination mechanisms do not work because of what the authors call differences in work place culture; this paper provides a language to express why such things happen, showing how the mismatch comes about, and what would be needed to address it. Using our theoretical framework, it is possible to explain that in Damian et al. [53] case, the Canadian and the US sub-teams were deploying different social protocols. While the Canadian team members relied on submitted messages, the US team members expected situated articulation messages through mail in order to be informed on the changes to the code. Thus, in the case described, the metawork was not successful, as it did not result in a shared coordination mechanism.

As another example, the conceptual framework allows to describe the media and the subject components of the "software informalisms" proposed by Scacchi [54]. The media side of a "software informalism" can be related to the communicative genre, while the subject is supported by coordination mechanisms. Therefore, e.g., a Forum message that contains the agenda of a meeting can be considered as a situated articulation genre (the media) as well as the artifact of the coordination mechanism (the subject). Scacchi considers Forum threads as software informalisms; thanks to our conceptual framework, it is possible to describe in greater detail how team members use the artifacts and how software informalisms can be developed and adopted in globally-distributed teams.

In summary, the conceptual framework allows the analyzing and describing of communicative and coordinative practices in distributed teams, and allows a better understanding of whether and how team members succeed in overcoming GSD challenges. In particular: (1) it allows to better understand existing research and provides researchers a language with which to describe practices and to express breakdowns; (2) it shows how GSD challenges can be overcome, e.g., adapting coordination mechanisms to obtain shared coordinative practices, and (3) it allows to show when distributed teams do not achieve a successful collaboration, e.g., when shared social protocols are not established.

Through the theoretical framework, it is also possible to explain that in GSD it is not sufficient to use the same set of tools in order to effectively collaborate remotely. Tools need to be adapted to a team's necessities and processes, and models need to be adopted by team members. This is further detailed in the next subsection.

8.2. About the Repertoire

The conceptual framework offers a way to describe whether and how team members succeed in adaptation and adoption of methods and tools, indicating how practices evolve over time through the evolution of social protocols. Coordination mechanisms and communicative genres are part of a repertoire (or ecology of channels [22]) that can be, and has to be, studied as a unitary system in order to understand the relationships between different channels and tools. The examples reported in this paper show that not one "right" set of coordination mechanisms and communicative genres can be identified, as different repertoires are adopted in different teams. However, based on the framework, a set of functions can be suggested that must be supported by the repertoire in order to achieve successful collaboration, such as, but not exclusively, metawork and socialization. Future research could investigate in detail what the additional needs are that have to be part of a repertoire to make effective.

8.3. Understanding the Role of Social Software

A literature review performed by the authors [10] has highlighted the lack of research papers focusing on understanding the relationship between SoSo and the ecology of tools used by globally-distributed teams. In this research work, SoSo appears as part of an ecology of channels [22] that has to be explored as a whole, not exclusively focusing on the specific functionality of each kind of SoSo. Thanks to the theoretical framework, the central role of SoSo in the teams analyzed is highlighted: on the one hand, it supports metawork to enact and negotiate coordination mechanisms; on the other hand, it supports different kinds of communicative genres, e.g., work discussions, knowledge sharing, articulation work, and team building. SoSo complements Software Engineering collaborative tools that provide a template of coordination mechanisms, supporting communication between team members. The analysis through the theoretical framework of different globally-distributed teams reported in this paper shows not only the dynamics of how social protocols are negotiated and established within novel distributed teams, but it also provides an indication of why breakdowns occur, describing how social protocols are maintained in an established team and highlighting the supportive role played by SoSo.

SoSo appears to be fundamental in the establishing phase of projects when decisions need to be taken, social relationships need to be established, and social protocols need to be negotiated. After the initial phase, Forum and Wiki serve as a persistent repository for the knowledge shared, while the communication through Forum and IM serve as a channel for situated articulation, decision making, and collaboration record, thus providing a channel for new metawork to adapt the existing coordination mechanisms further. IM is the media where things happen if they do not have an established place anywhere else. IM, therefore, also serves as a dispatcher for other channels and as the main channel for social talk. SoSo enables social talk and metawork, both necessary for establishing and maintaining successful collaboration: social talk supports the actual work in distributed environments by enhancing the development of social relationships between team members. Very importantly, metawork is fundamental for the software development activities, the negotiation and establishment of social protocols in the collaboration. Providing to distributed teams the access to flexible tools such as SoSo and encouraging team building, chats appear to have a positive impact on the success of the collaboration.

8.4. Implications for Research and for Practice

This paper has shown how situated action [13] can be analyzed to study communicative and coordinative practices in globally-distributed teams. Practices in situated action are not fully specified by software engineering methods and processes. Thus, adaptation by teams is necessary. The concept of social protocols as part of both communicative genres and coordination mechanisms helps to further explore these processes. A theoretical framework based on the notion of social protocols is useful to understand breakdowns, to investigate the establishment of practices in novel distributed teams, and to analyze the re-negotiations of practices in established teams. The importance of negotiating and agreeing on common social protocols is particularly decisive in GSD settings in which direct communication cannot easily take place, as it is often mediated by artifacts in which socio-cultural distance can affect the collaboration. Thus, agreeing on social protocols becomes even more challenging and crucial. The theoretical framework can be beneficial for future research that aims to analyze and describe not only the role of SoSo, but also how communicative and coordinative practices are established and maintained in distributed teams.

An encouraging contribution, both for researchers and practitioners, is that the success of the collaboration strongly relates to how coordination mechanisms and social protocols are established between team members, thus helping to bridge language barriers, cultural differences, as well as time and space distance. The analysis indicates the importance of communication supported by SoSo as a side channel that complements professional coordination mechanisms. It does not appear necessary to design new tools; nevertheless, it is very important to understand how existing tools are adopted and what the potentials are of their usage. Channels other than SoSo can certainly be used; however, it seems advisable for practitioners to ensure that metawork and social talk are supported by tools used by team members and that they actually take place, as geographical distribution does not preclude the possibility of having these kinds of conversations.

8.5. Limitations and Future Work

The theoretical framework allows the understanding of the heterogeneous use of channels and artifacts in two very different cases. The two cases have been investigated in detail in two previous articles [22], [23]. In this paper, both the industrial and the student projects have been employed to show that the framework allows to describe the actual practices of practitioners in an established team, as well as the establishing and maintaining of practices in a newly formed team composed by novice developers.

However, the framework does not give an explanation as to why, for example, in the student projects, two out of three teams did not perform well. It does, however, allow to describe the different characteristics of the various collaborations, and to show how the two teams did not perform, e.g., not communicating through SoSo, not having metawork, and not establishing effective coordination mechanisms. This indicates that having different teams with different characteristics does not impact on the trustworthiness of the framework; rather, it shows that the framework is able to provide a better indication of how the collaboration can fail. Combining an analysis of the repertoire of communicative genres and the set of coordination mechanism would allow to more clearly characterize different types of teams, e.g., tightly collaborating teams or loosely-coupled teams. One would expect that such different teams show different characteristic patterns in their repertoire. The framework could potentially also be used to analyze co-located communicative and coordinative practices in sub-teams; however, this aspect is beyond the scope of this study, which has focused on communication through SoSo within the entire distributed team. However, face-to-face communication could also be analyzed to investigate, through genre analysis of video recorded meetings, the co-located communicative practices in different locations. This could lead, for example, to describing the importance of metawork in co-located settings, as highlighted by Rönkkö et al. [55] in the analysis of co-located steering group meetings.

Another limitation is that the analysis of the collaboration within the team does not aim to be exhaustive: further genres could be identified and genres categories, other than the ones analyzed in the present paper, could occur in different media. For example, virtual meetings could have been recorded and likewise analyzed, adapting the communicative genre's analytic concept. Moreover, the coordination mechanisms described in this article were mostly established and persistent during the collaboration; however, it could be possible to identify further temporary coordination mechanisms established by team members for specific purposes, driven by particular situations occurring during the collaboration. Thus, further communicative and coordinative practices could be included when examining further cases through the framework.

Finally, the theoretical framework focuses on communicative and coordinative practices in distributed collaboration: common practices adopted by globally-distributed software teams definitely also include other kinds of practices, such as coding practices or design practices. We kept the focus on communicative and coordinative practices, as they allow understanding the role of SoSo in the cooperation mediated by artifacts. Thanks to the framework proposed, it is possible to investigate and understand the role of SoSo within the repertoire used in the projects studied. However, future research could extend the framework, including additional theoretical concepts for analyzing and describing further kinds of practices.

9. Conclusions

This paper presents a novel framework to analyze and describe coordinative and communicative practices in Global Software Development (GSD) based on the concepts of communicative genres and coordination mechanisms. The conceptual framework allows researchers to describe and analyze computer-mediated collaborative practices in distributed settings, highlighting the importance of social protocols. It allows a better understanding of existing research, the heterogeneity of practices in globally-distributed teams, and the role of Social Software (SoSo) within the repertoire used in the teams. Through the framework, it is possible not only to highlight that SoSo allows team members to establish, develop, and maintain social protocols during the collaboration, but it also supports metawork and team building chats. The theoretical framework can be beneficial for future research that aims to analyze and describe not only the role of SoSo, but also how cooperative practices are established and maintained in globally-distributed software teams.

10. Acknowledgments

The authors would like to thank Dr. Irina Shklovski for the insightful comments and feedback on an earlier version of this article. Moreover, the authors would like to thank Dr. Paolo Tell for the support with the coding of the field material, for indispensable discussions on the topic and for the always-on research support.

Appendix A. Interview guideline example (Interview with the DHI project manager)

- Why did you choose to use Spira Team? Who did take the decision? Did you evaluate other tools? How did you do before?
- Do you use the History functionality available in Spira to reconstruct a task or an incident?
- How do you know what people are exactly working on? Can you show it to me on Spira? Would you find useful knowing exactly what Indian developers are working on?
- How do you see in Spira when a task/incident status has been changed? Are you just looking at what is assigned to you? Would it be good to have kinds of "story" of what the developer did?
- How did you start using Skype? Why not Messenger or other jabber chat?
- Do you ever read your Skype logs history? Why?
- Do you usually have a meeting in Copenhagen every day from 10 to 11? If not, why? And what does it happen if someone is working from home?
- Who is responsible to approve/plan incidents? Do you usually do it only during the meeting or can you do it also on your own?
- How do you distribute the work among Indian developers?
- An incident detected by a PAM or a developer, after the developer fixes it, is assigned again to the person that detected the incident and not to the Indian tester. Is that right? Is it because there is no test case related to it? If the fixing of the incident creates a bug, the Indian tester will discover it in the next cycle of test, right?
- I did not get what is the purpose of the spreadsheet that the team leader does in India. Can you explain it to me?

Appendix B. Coding Schema

Coding of the Purpose of Communicative Genres:

- Knowledge Sharing
- Team building
  - Encouraging Chats
  - Socialization
- Articulation Work
  - Metawork
  - Situated Articulation
- Work

Coding of the dimensions within each Purpose of Communicative Genres for defining sub-genres:

- Response
- Question
- For Your Information (FYI)
- Proposal
- Meta-comment

Appendix C. References

[1] Y. Dittrich, D. W. Randall, J. Singer, Software engineering as cooperative work, in: Computer Supported Cooperative Work (CSCW), Vol. 18, Springer, 2009, pp. 393–399.

[2] K. Schmidt, L. Bannon, Taking cscw seriously, in: Computer Supported Cooperative Work (CSCW), Vol. 1, Springer, 1992, pp. 7–40.

[3] K. Schmidt, C. Simone, Coordination mechanisms: Towards a conceptual foundation of cscw systems design, in: Computer Supported Cooperative Work (CSCW), Vol. 5, Springer, 1996, pp. 155–200.

[4] J. D. Herbsleb, Global software engineering: The future of socio-technical coordination, in: 2007 Future of Software Engineering, IEEE Computer Society, 2007, pp. 188–198.

[5] E. Carmel, R. Agarwal, Tactical approaches for alleviating distance in global software development, in: Software, IEEE, Vol. 18, IEEE, 2001, pp. 22–29.

[6] J. D. Herbsleb, D. Moitra, Global software development, in: Software, IEEE, Vol. 18, IEEE, 2001, pp. 16–20.

[7] M. Cataldo, M. Bass, J. D. Herbsleb, L. Bass, On coordination mechanisms in global software development, in: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, IEEE, 2007, pp. 71–80.

[8] H. Holmström, B. Fitzgerald, P. J. Ågerfalk, E. Ó. Conchúir, Agile practices reduce distance in global software development, in: Information Systems Management, Vol. 23, Taylor & Francis, 2006, pp. 7–18.

[9] A. M. Kaplan, M. Haenlein, Users of the world, unite! the challenges and opportunities of social media, in: Business horizons, Vol. 53, Elsevier, 2010, pp. 59–68.

[10] R. Giuffrida, Y. Dittrich, Empirical studies on the use of social software in global software development-a systematic mapping study, in: Information and Software Technology, Elsevier, 2013.

[11] F. Lanubile, Collaboration in distributed software development, in: Software Engineering, Springer, 2009.

[12] Y. Dittrich, C. Floyd, R. Klischewski, Social thinking, software practice, The MIT Press, 2002.

[13] L. A. Suchman, Plans and situated actions: the problem of human-machine communication, Cambridge university press, 1987.

[14] K. Kautz, J. Nørbjerg, Persistent problems in information systems development. the case of the world wide web., in: ECIS, 2003, pp. 919–926.

[15] G. Avram, Of deadlocks and peopleware-collaborative work practices in global software development, in: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, IEEE, 2007, pp. 91–102.

[16] P. A. Nielsen, J. Nørbjerg, Assessing software processes: low maturity or sensible practice, in: Scandinavian Journal of Information Systems, Vol. 13, 2001, p. 5.

[17] A. Sigfridsson, A. Sheehan, On qualitative methodologies and dispersed communities: Reflections on the process of investigating an open source community, in: Information and Software Technology, Vol. 53, Elsevier, 2011, pp. 981–993.

[18] A. Boden, G. Avram, L. Bannon, V. Wulf, Knowledge management in distributed software development teams-does culture matter?, in: Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on, IEEE, 2009, pp. 18–27.

[19] J. Singer, T. Lethbridge, N. Vinson, N. Anquetil, et al., An examination of software engineering work practices, 1997.

[20] D. Nicolini, S. Gherardi, D. Yanow, Knowing in organizations: A practice-based approach, ME Sharpe, 2003.

[21] A. Boden, B. Nett, V. Wulf, Coordination practices in distributed software development of small enterprises, in: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, IEEE, 2007, pp. 235–246.

[22] Y. Dittrich, R. Giuffrida, Exploring the role of instant messaging in a global software development project, in: Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on, IEEE, 2011, pp. 103–112.

[23] R. Giuffrida, Y. Dittrich, How social software supports cooperative practices in a globally distributed software project, in: Cooperative and Human Aspects of Software Engineering (CHASE), 2014 7th International Workshop on, ACM, 2014.

[24] W. J. Orlikowski, J. Yates, Genre repertoire: The structuring of communicative practices in organizations, in: Administrative science quarterly, JSTOR, 1994, pp. 541–574.

[25] E. Ó. Conchúir, P. J. Ågerfalk, H. H. Olsson, B. Fitzgerald, Global software development: where are the benefits?, in: Communications of the ACM, Vol. 52, ACM, 2009, pp. 127–131.

[26] D. Šmite, C. Wohlin, Z. Galviņa, R. Prikladnicki, An empirically based terminology and taxonomy for global software engineering, Empirical Software Engineering 19 (1) (2014) 105–153.

[27] H.-K. Cho, M. Trier, E. Kim, The use of instant messaging in working relationship development: A case study, in: Journal of Computer-Mediated Communication, Vol. 10, Wiley Online Library, 2005, pp. 00–00.

[28] T. Niinimaki, C. Lassenius, Experiences of instant messaging in global software development projects: A multiple case study, in: Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on, 2008, pp. 55 –64.

[29] T. LaToza, G. Venolia, R. DeLine, Maintaining mental models: a study of developer work habits, in: ICSE '06: Proceedings of the 28th international conference on Software engineering, 2006.

[30] M. Raman, Wiki technology as a" free" collaborative tool within an organizational setting, in: Information systems management, Vol. 23, Taylor & Francis, 2006, pp. 59–66.

[31] J. Zhang, Y. Qu, J. Cody, Y. Wu, A case study of micro-blogging in the enterprise: use, value, and related issues, in: CHI '10: Proceedings of the 28th international conference on Human factors in computing systems, 2010.

[32] M. Storey, J. Ryall, J. Singer, D. Myers, L.-T. Cheng, M. Muller, How software developers use tagging to support reminding and refinding, in: Software Engineering, IEEE Transactions on, Vol. 35, IEEE, 2009, pp. 470–483.

[33] C. Treude, M. Storey, How tagging helps bridge the gap between social and technical aspects in software development, in: Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on, IEEE, 2009, pp. 12–22.

[34] G. A. Okhuysen, B. A. Bechky, 10 coordination in organizations: An integrative perspective, in: The Academy of Management Annals, Vol. 3, Taylor & Francis, 2009, pp. 463–502.

[35] T. W. Malone, K. Crowston, The interdisciplinary study of coordination, in: Computing Surveys, Vol. 26, ACM, 1994, p. 87–119.

[36] K. Crowston, H. Annabi, J. Howison, C. Masango, Effective work practices for software engineering: free/libre open source software development, in: Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research, ACM, 2004, pp. 18–26.

[37] J. Whitehead, Collaboration in software engineering: A roadmap, in: Future of Software Engineering, 2007. FOSE'07, IEEE, 2007, pp. 214–225.

[38] D. E. Strode, S. L. Huff, B. Hope, S. Link, Coordination in co-located agile software development projects, in: Journal of Systems and Software, Vol. 85, Elsevier, 2012, pp. 1222–1238.

[39] I. R. McChesney, S. Gallagher, Communication and co-ordination practices in software engineering projects, in: Information and Software Technology, Vol. 46, Elsevier, 2004, pp. 473–489.

[40] B. Jordan, A. Henderson, Interaction analysis: Foundations and practice, in: The journal of the learning sciences, Vol. 4, Taylor & Francis, 1995, pp. 39–103.

[41] C. Robson, Real world research: a resource for social scientists and practitioner-researchers, Vol. 2, Blackwell Oxford, 2002.

[42] D. Chandler, An introduction to genre theory, in: The Media and Communications Studies Site, 1997.

[43] J. Yates, W. J. Orlikowski, Genres of organizational communication: A structurational approach to studying communication and media, in: Academy of management review, Vol. 17, Academy of Management, 1992, pp. 299–326.

[44] C. R. Miller, Genre as social action, in: Quarterly journal of speech, Vol. 70, Taylor & Francis, 1984, pp. 151–167.

[45] H.-G. Im, J. Yates, W. Orlikowski, Temporal coordination through communication: using genres in a virtual start-up organization, in: Information Technology & People, Vol. 18, Emerald Group Publishing Limited, 2005, pp. 89–119.

[46] Y. Wang, D. Redmiles, Understanding cheap talk and the emergence of trust in global software engineering: An evolutionary game theory perspective, in: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on.

[47] E. M. Gerson, Reach, bracket, and the limits of rationalized coordination: Some challenges for cscw, in: Resources, Co-Evolution and Artifacts, Springer, 2008, pp. 193–220.

[48] A. Strauss, The articulation of project work: An organizational process, in: The Sociological Quarterly, Vol. 29, Wiley Online Library, 1988, pp. 163–178.

[49] L. Pries-Heje, Four different paradigms for process design when implementing standard enterprise systems, in: CONFENIS 2010, 2010.

[50] L. Wittgenstein, Philosophical investigations, Oxford: Blackwell, 1953.

[51] M. Yasuoka, Bridging and breakdowns, Ph.D. thesis, Copenhagen Business School (2009).

[52] S. R. Barley, The alignment of technology and structure through roles and networks, in: Administrative science quarterly, JSTOR, 1990, pp. 61–103.

[53] D. Damian, L. Izquierdo, J. Singer, I. Kwan, Awareness in the wild: Why communication breakdowns occur, in: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on, IEEE, 2007, pp. 81–90.

[54] W. Scacchi, Free/open source software development: Recent research results and methods, in: Advances in Computers, Vol. 69, Elsevier, 2007, pp. 243–295.

[55] K. Rönkkö, Y. Dittrich, D. Randall, When plans do not work out: How plans are used in software development projects, in: Computer Supported Cooperative Work (CSCW), Vol. 14, Springer, 2005, pp. 433–468.