Exploring the Role of Instant Messaging in a Global Software Development Project

Yvonne Dittrich  
Software Development Group  
IT University of Copenhagen  
Copenhagen, Denmark  
ydi@itu.dk

Rosalba Giuffrida  
Software Development Group  
IT University of Copenhagen  
Copenhagen, Denmark  
rogi@itu.dk

Abstract— Communication plays a vital role in software development projects. Globally distributed teams use a mix of different communication channels to get the work done. In this paper, we report on an empirical study of a team distributed across Denmark and India. This paper explores the integration of formal documentation, bug-tracking systems and email with informal communication on Instant Messaging (IM), screen sharing, and audio conversations. Whenever overlap times occur, informal communication can take place at the same time in different sites, and it can effectively complement formal documentation. Our analysis provides an indication that IM can play a special role in such socio-technical communication systems: IM acts as a real time glue between different channels. The communication through IM also provides a means to build trust and social relationships with co-workers.

Instant messaging, skype, social software, informal communication, ethnographic research.

I. INTRODUCTION

Software development teams are used to working with many different tools for developing software, versioning it, scheduling work, managing requirements and test cases, and sharing knowledge with other team members. There are several tools available for these activities, such as: integrated development environments, versioning systems, bug-trackers, test management tools, word documents and email. These tools can be considered as channels in which formal communication takes place among virtual teams.

Formal communication is not always sufficient for conducting successful software projects. In co-located teams, planned or impromptu face-to-face meetings can easily support the development process, complementing the formal knowledge being shared through the use of tools [1]. In distributed teams, informal communication cannot take place in person very easily, and thus, other kinds of unstructured channels are used in Global Software Development (GSD), e.g. IM, voice calls, and screen sharing. Turner et al. [34] have explored the workplace communication ecology, investigating why particular tools in a user’s ecology are selected, and how the tools are used in combination.

In this study, we investigate the concept of communication ecology in globally distributed teams in order to answer the following question:

How do software distributed teams manage communication using different channels and how do they integrate the communication across different channels?

In our analysis, we refer to the concept of coordination mechanisms proposed by Schmidt and Simone [25]: coordination mechanisms consist of a coordinative social protocol sometimes imprinted on, and supported by, distinct artifacts. They stipulate and mediate the articulation of cooperative work so as to reduce the complexity of the articulation work of that arrangement. A bug tracker, for example, collects bug reports and supports structured asynchronous communication that sustains the debugging process. Skype™ is an example of less structured communication, nearly a synchronous communication channel. Coordination mechanisms [25] can use structured and unstructured communication channels. For unstructured channels, however, it is necessary to re-negotiate coordination mechanisms [25].

This paper reports a qualitative study on collaboration in a distributed software development team. We observe that IM plays a crucial role in supporting daily work and coordination across sites: IM acts as the glue between different channels available in the communication ecology [34] of the virtual team.

We believe that our study can help to bring about a better understanding of how formal and informal communication can be combined in GSD, as well as to understand the crucial role of IM for coordination, collaboration and socialization in software distributed teams.

II. RELATED LITERATURE

The fundamental problem of GSD is that many of the mechanisms that function to coordinate the work in a co-located setting are absent or disrupted in a distributed project [14]. Bruegge et al. [3] have proposed a framework for enabling informal collaboration in global software development, underlying the crucial role that informal mechanisms play in distributed software teams.

While traditionally the main media for informal communication in distributed teams have been email, phone, and video conferencing systems [1] [11], nowadays a lot of
communication takes place in the so-called Social Software (SoSo) [10]. SoSo can be seen as a repository of non-structured knowledge where informal communication and knowledge sharing take place. SoSo includes a wide variety of tools such as: IM, Internet forums, mailing lists, blogs, wikis, social network sites, social bookmarking, social libraries, and virtual worlds. The main challenge from a Software Engineering perspective is that informal communication is not structured by itself. While Software Engineering usually focuses on structured documentation and communication, knowledge shared through SoSo is unstructured. As high-expertise practitioners involved in distributed team projects share their knowledge both through formal documentation and informal situated meanings, SoSo should be taken into account as a complementary link.

Many research projects in the CHI, CSCW and IS communities have already investigated the applications of SoSo in corporate contexts [32] [33] [35]. Despite the increase of Web 2.0 tools, research on the use of SoSo in workplaces still focuses mainly on IM. One of the first papers on IM for collaborative work was written in 2000 by Nardi et al. [20]; the authors document the flexibility and expressivity of IM for various informal communication tasks in the workplace. In 2010, Ou et al. [22] demonstrated that IM can empower teams at work. In the last ten years, a lot of research has been conducted on the use of IM in the workplace, especially in distributed settings; we have identified three main empirical research directions: evaluation of self-made IM tools, laboratories studies with students, and analysis of practices adopted in open source communities.

Hubbub and WebWho are two examples of IM tools developed in the early 2000s. Issacs et al. [17] logged thousands of workplace conversations among users of their Hubbub system, and evaluated the nature and functions of the conversations. They found that the primary use of Hubbub was for work conversations; secondary uses were for simple, single purpose interactions and for scheduling or coordination. Studies conducted with WebWho, a web based awareness system that visualizes where people are located in a large university computer lab, show that students use the IM to support collaborative work and coordinate social activities, and use it extensively for playful behavior [27].

The laboratory studies conducted by Setlock et al. [28] mainly underlined the importance of cultural aspects in computer mediated communication. Other ad-hoc studies have investigated the effects on interpersonal perception [6] or the effects on cooperation, persuasion, and deception [2] or comparisons between chat and audio in media rich environments [26]. Also, in OSS settings, the use of IM has been researched extensively. Elliot [9] indicates the importance of recorded logs of IM for resolving conflicts in virtual work communities. The use of persistent online IRC logs and mailing list archives serves to tie the virtual work group together, contributing to conflict resolution and, at the same time, reinforcing the beliefs in free software and freedom of choice, as well as the more tacit values of cooperative work and community building. Gutwin et al. [12] affirm that text chat is suitable for ad-hoc communication and ‘overhearing’ of informal and work-related discussions. We notice that little empirical research on the use of IM in GSD has been reported, and thus believe it is appropriate to investigate this aspect in real world settings.

Niinimaki and Lassenius [21] have studied the use of commercial-grade, widely adopted and acknowledged IM tools in commercial GSD projects, through a multiple qualitative case study. They reported on successful use of IM in global distributed projects. Thissen et al. [31] describe the role of different communication tools for distributed software development teams, and Turner et al. investigate [34] how people set their own communication ecology in the workplace according to communication needs. Combining these two works, we conclude that in GSD a team’s communication ecology of different tools used for different purposes is evident.

Based on ethnographic research, we emphasize the central role of IM as glue between different channels mediating the collaboration and assemblies of boundary objects [4] [19]. We analyze the role of IM by identifying and investigating different dimensions of chat usage. Handel and Herbsleb [13] have already built a classification of chats, defining five categories: availability, non-work topics, work, greeting, and humor. In our analysis, we observe that, probably because today IM is more integrated in everyday work practices, all chats are work related and several dimensions are present at the same time.

III. CASE DESCRIPTION AND METHOD

DHI is an independent, international, consulting and research organization. The company develops and uses high-end hydraulic simulation software. We have investigated how informal communication takes place inside the World Bank Project (WB-Project), which has a considerable amount of ongoing software development. This is a global distributed project: five members are in Copenhagen, Denmark; seven members in Delhi, India and one Project Area Manager in Portland, USA. The Danish team is composed of one project manager and four Project Area Managers (PAM), the Indian team consists of five developers, one team leader, and one tester. Each PAM is responsible for one specific part of the software and collaborates with one or two Indian developers. Project Area Managers are both domain experts of hydraulic engineering and software developers. They take care of the management of the project, requirement specifications, quality assurance processes as well as the design of the software and its implementation. 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. The project was successful, and the team is now developing the second release of the software.

The ethnographic empirical research took place both in Copenhagen and in Delhi. Researchers collected field material and observed the team for four months from
Denmark. In order to have a better understanding of the practices, researchers also spent two weeks in India.

For this study, researchers collected data through different qualitative empirical techniques: participant observation, semi-structured interviews, member checking and document analysis. In order to carefully track the investigation, researchers taped and transcribed meetings and interviews, they applied interaction analysis on IM logs, and they kept a research diary during the entire project. Workshops were organized to summarize the outcomes and to support researchers and practitioners in reflecting together on observations. By participating in the daily routines, it was possible to observe how collaboration took place between different participants of the development project. Interviews gave the opportunity to clarify uncertainties and for interviewers to ask about specific issues in a deeper way.

The analysis of the data was initially performed during the observation period through several iteration of analysis of the field material. 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.

By using multiple ways of collecting data and combining different kinds of methods, it has been possible to triangulate the findings [24]. The goal of the empirical part of the research is to understand and support software development from a shop floor perspective, as proposed by Dittrich et al. [8] in the Cooperative Method Development (CMD), which is an approach for investigating social and cooperative aspects of software development. With the support of the researchers, team members had the opportunity to reflect on their own practices and to understand difficulties that they encountered during everyday collaboration.

IV. ANALYSIS

In this section, we describe the WB-Project team, outline which tools the team used, determine how they were used, and present the incident workflow. Finally, we describe the different dimensions of the use of IM that play a central role in the communication ecology of the team.

A. The World Bank project team

The working day of the team starts in India. In the Delhi office, developers arrive starting from 9 a.m., Indian Standard Time (IST) or 4:30 a.m., Central European Time (CET). They are gathered all together in an open space. 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. The morning is rather quiet, as developers finish their work of the previous day or start newly assigned tasks. Sometimes they move to another workspace to communicate with other team members.

Collaboration across sites takes place during four to six overlapping working hours for different purposes: clarifying requirements, debugging pieces of software, and coordinating work tasks. Real time collaboration and communication across sites usually starts through Skype™ instant messaging, but a whole set of channels is used by team members, depending on the specific task that they have to perform.

At the time of the late Indian morning, Danish managers wake up with some of them starting to work around 8 a.m., CET or 12:30 IST, often from home. When Danish team members appear online on Skype™, Indian developers start asking them questions, or a Danish PAM might call Indian developers to organize his working day. Some real-time collaboration can take place in this short slot before the Indian lunch that takes place at 1 p.m., IST. Collaboration is usually interrupted at the Indian side for lunch hour, as all team members take lunch together.

While the Indian team is having lunch, the Danish team members reach the office. The five team members are spread in four two-person-offices in two different floors. After Indian members come back from lunch, at 10 a.m., CET, Danish team members usually have a meeting. During this routine meeting, PAMs plan future activities, prioritize tasks and assign incidents and tasks to team members. The meeting usually lasts about one hour. The time after the meeting, from 11 a.m. to 12 a.m., is normally one of the most intense times for communication across sites.

About 12 noon, CET, the Danes have lunch. Their lunchtime is flexible and depends on their current activities, as they usually do not share lunch. When the Danish team is back from lunch, the working day in India is close to ending. The last slot for real time communication and collaboration is from 5 p.m. to 6 p.m., IST. The rest of the Danish afternoon is quieter when local collaboration and meetings take place.

B. Different tools used

The main structured collaboration support is Spira™, a Test Management tool from Inflectra Corporation. It is used by the team as an issue tracking system. All development activities are tracked in this system: the description of features to develop, reporting of incidents, assignment of tasks, and description of test cases. Spira™ is a test case tool rather than a project management tool, but it is used by the team to manage tasks of the team members. The current version has some limitations for user management and there are few bugs in the activity reporting functionality. Spira™ automatically assigns an “incident number” (IN) to all defects, test cases (TC) and requirements (REQ). These numbers are used by the team in all tools to link the Spira™ entries with chat, mail, documentation and source code: for example, every time developers check in code in the source repository, they also report the related number of Spira™ (See Figure 1).

Skype™ is the team’s main tool for collaboration and team members are supposed to have it switched on 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 communication and collaboration take place through Skype™. Through the contact list, team members get an overview of the availability of remote colleagues. The availability information in the contact list provides a sense of connectedness, bringing people together.
Despite the fact that the screen and they use a proprietary tool developed by DHI that leads to delay point and manipulate the software and tools, only one side can rather than the other way round. Skype™, however, does not observed that developers review and also for short session of pair programming. We screen sharing is used for debugging, for code/solution communication availability of the colleague. IM happen only if the person is online on that contact. In other cases, they prefer using mails. Danish team members use IM for checking urgent or high priority issues. Indian developers state that they contact Danish team members through Skype™ whenever immediate feedback is required or due to urgent or high priority issues. In other cases, they prefer using mails. Danish team members use IM for checking what developers are working on, as well as for giving clarification and feedback. The function of chat is further analyzed in Section D of the analysis.

Voice over IP allows team members to talk together and is replacing traditional phone connections. Due to the fact that contact availability can be seen on the contact list, calls happen only if the person is online on Skype™. Moreover, the call is usually negotiated beforehand, through a chat on IM, so that each member of the team can check the actual communication availability of the colleague.

While talking to each other, team members often share screen. Referring to the same indexical space, team members can discuss code, software behavior, or incident description. Screen sharing is used for debugging, for code/solution review and also for short session of pair programming. We observed that developers usually show their screen to PAMs, rather than the other way round. Skype™, however, does not support remote control of the desktop. Only one side can point and manipulate the software and tools, a limitation which leads to delays and misunderstandings.

When team members need to control the remote desktop, they use a proprietary tool developed by DHI that is mainly used for remote presentations for customers. This tool requires preparation by the person who wants to share the screen and the sending of the code to the remote colleague. The decision to use this facility is usually made during a call, when the Skype™ screen sharing turns out to be insufficient. Despite the fact that video is one of the core features of Skype™, one important factor that we observed is that no one in the team uses video. In the WB-project the team does not miss face-to-face contact at all. The following citation is an example of what both the Indian developers and Danish team members reported:

I don’t have a camera but I can use the external one if I want. I don’t feel we need to use video... Actually we use video: we share our screen when necessary.

Together with Spira™ and Skype™, team members also make use of traditional email. Emails are exchanged among all team members, mainly when there will be a time delay for finishing a solution or when it is necessary to involve more people in the discussion. Especially when the input from the US team member is necessary, the communication takes place by using emails as there is only little overlap in working hours with Portland. Email is also used to broadcast information, to formalize decisions, and to ask for more formal response.

Both Word and PDF documentation are used to formalize the planning and the execution of the WB project. Project Organization and Project Life Cycle are formally defined in a document with guidelines for the Software Development Project. A document incorporates a short, step-by-step guide on how Spira™ should be used by the team to manage software development projects. These documents are shared through mail or are uploaded in Spira™. Interestingly, in these documents, Skype™ is not mentioned as a tool for supporting the software development process.

C. The incident work flow

In the “Project handbook” document, there is a diagram that formally describes the workflow of a defect (see Figure 2). The defined life cycle is roughly followed. 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 managers – Change Control Board (CCB) - usually approve and plan the incident, assigning it to a developer through Spira™. When the developer starts working on the defect, he changes its status as “in progress,” and when he finishes working on that, he marks it as “completed.” He then assigns the incident to the PAM, who can test and verify it. This process, defined through the diagram, is strictly mapped in the Spira™ system, and the status of all defects is also structured in the tool. Apart from defect definition and design comments in the Spira™ through the whole workflow, a lot of informal communication takes place in order to clarify the formal specification. Mails, voice calls, screen sharing and IM chats enrich the Spira™ content.

Following the lifecycle of a specific incident, we identify three phases in which informal communication usually takes place across sites: when the defect is assigned to the PAM, when the developer works on the defect, and when the defect is in status “completed.”
In the first phase, 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, he chats with the Indian member through IM, or he may ask to start a screen sharing session in order to better understand the problem. If a developer is working on the resolution of an incident, and has doubts on how to implement the solution described in Spira™, he contacts the PAM to discuss the technical issues through IM, audio, or screen sharing. Finally, when an incident is closed, 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 first case, the PAM uses chats and audio to talk directly with the developer. In the second case a mail is sent to all the PAMs and developers involved, in order to inform all members about the problem and to find a shared solution. In other words, the resolution of incidents takes place not only through Spira™ and through the source code repository, but deploys a whole range of structured and unstructured channels. The informal communication and collaboration makes the formal process work.

D. Dimensions of the use of IM

In this section, we focus on informal communication through IM and we analyze and classify the chat content. The different dimensions that we consider for our analysis are: coordination, collaboration, awareness, and socialization. Each chat can normally be classified under a predominant dimension. Interestingly, often more than one dimension can come into play at the same time and in the same piece of chat. We here briefly describe the four dimensions, before we show how they become visible in the different chats.

The coordination dimension is the most prominent one. Although tasks and incidents are assigned to team members through Spira™, it often happens that people coordinate informally on what to do next or on managing the lifecycle of an incident or a task. Chats are another consideration for negotiating a call as a coordination chat. Usually team members do not call directly people on their contact list; the availability of the counterpart is checked through a written chat. The call is often required when people need to discuss a specific task or incident or want to show/check solutions implemented.

When technical issues are discussed the collaboration dimension becomes visible. These chats are usually easy-to-resolve issues that do not require much discussion. We observed that collaboration chats usually last for no more than 10 minutes; on average, they last 7 minutes. When the topic is too complex for a written chat, the conversation started in IM usually switches to a call. Moreover, if issues are considered to have a wider implication, developers or PAMs may start a discussion on mail.

For the awareness dimension chats are used: for example, Indian developers may mention what they are working on in order to avoid conflict. On the other hand, Danish managers check explicitly on what people are working on to plan future activities. This information is provided by Spira™, but often what the official tool shows will not necessarily correspond to what people are really working on at the moment. This is due to technical limitations of the tool and also because of remiss reporting of activities. Many of the coordination chats provide awareness information to the cooperation partner.

The socialization dimension is visible both in dedicated social chats and as sub-text in other chats. They are usually talks unrelated to code regarding working conditions, commuting problems or personal topics. These chats can be longer than coordination and collaboration ones but, since they are low priority conversations, usually they are going on while people are also working on some other tasks. The socialization dimension is sometimes shown as an addendum to a coordination or awareness chat, e.g., as positive feedback for the good job. This kind of chats is useful for team building and mutual appreciation, as well as for motivation. During many collaboration and coordination chats, implicit exchange of awareness and socialization as sub-text takes place and it is very interesting to see how these aspects are interwined together with coordination and collaboration activities.
1) The Coordination Dimension

Despite the fact that all team members have to assign tasks or incidents to other colleagues on Spira™, we observed that many coordination chats start from the Indian side to inform the task/incident changed status, and is assigned to the counterpart.

[14:07:17] IND: 974 is yours now for verification (gap filler scrollbars)
[14:09:47] DK: thanks

(1)


(2)

As we can see in the previous example, coordination chats give implicit awareness to Danish PAMs about what developers are working on, and about what they are supposed to do next, such as checking the solution implemented.

We observe frequent coordination from both sides in very close exchange of assignments, as in pair programming settings:

[14:07:17] IND: 974 is yours now for verification (gap filler scrollbars)
[14:09:47] DK: thanks
[12:07:32] DK: all yours
[15:58:02] DK: 847 is yours

(3)

It happens also that developers ask what the tasks are that they are supposed to do, as in the following example:

[08:24:46] IND: any incident you have planned for me...
[08:24:57] DK: to fix
[08:28:15] DK: look in SPIRA for planned - take from the top with respect to priority

(4)

Similarly, chat is used to communicate about sending/receiving mail:

[20:26:27] DK2: writing ou an email now :) 
[20:26:55] DK1: just replied to it
[20:30:14] DK2: no...another one :)
[20:30:17] DK2: just sent it now

(5)

[07:28:33] IND: Hi DK...I have sent a mail..pls check
[07:29:57] DK: ok

(6)

Also in this case the coordination dimension is intertwined with the socialization one. We can observed a need to give attention to specific artifacts that are considered crucial for developing the work.

Negotiation chats look pretty similar, whether started from the Indian or Danish side; however, it is more typical that they are started by the Indian developers. Two examples, from both sides, are reported below.

[10:51:52] IND: hi..
[10:59:05] DK: hi
[10:59:49] IND: can we discuss in811
[11:10:58] ** Call to IND, no answer. **
[11:11:16] DK: I'll call again in 10 minutes - need to talk to somebody

(7)

[11:14:22] IND: ok, sorry i went to IND2 as he had some problems in the build
[11:15:28] DK: let’s talk now - I never got away from the keyboard :)
[11:16:13] IND: he he.. ok :
[11:16:47] ** Call from IND, duration 10:42. **

[09:40:59] IND: I have implemented […]
[09:41:38] IND: the implementation is working fine
[09:41:58] IND: do u want me to check-in the code and test on ur system??

(8)

Interestingly, it is only the first call of the day that is negotiated so politely and verbosely. When other calls during the day occur they are either a direct call - without checking availability of the other - or some fast negotiation such as:

[15:28:16] IND: Hi DK...r u there?
[15:28:37] DK: yes

[15:28:53] ** Call to DK, duration 08:59. **

(9)

2) The Collaboration Dimension

Collaboration chats are technical discussions used not only whenever developers have doubts on how to develop something or on what an incident means, but also when they need clarification on what the software should do. Danish managers start collaboration chats with developers whenever they want to understand what a specific piece of software does or when they need to know the implementation choices of the developers.

[13:35:51] DK: 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 DK wanted hidden, not just the standard "ID" field.

[13:36:49] DK: Yes, I think the criteria should be if the field type is "Guid", not what the field name is.
[13:36:49] IND: but can some other guid fields be important.
[13:37:07] DK: Possibly, but for now we want them all hidden.

Collaboration chats are less frequent than coordination chats, but they are definitely precious discussions in which design decisions are taken. For this reason these chats could be stored in a persistent channel. As reported by one practitioner:

*In theory, you should write down in a design document all the things that you discuss. We do that, but we don't do for Skype conversations. We could attach the log to an incident.*

Collaboration chats complement formal design specifications; even if requirements are not very well specified, coordination chats help clarifying what an incident means, how to develop a specific piece of software and what the software should do under specific conditions. As observed, usually collaboration chats do not last more than 10 minutes, and thus technical issues are solved with very fast interactions between co-workers, probably faster than detailing formal design documents. Interestingly, managers agree on the fact that sometimes it is easier to have a direct discussion about a specification rather than detailing it formally. On the other hand, they do state:

*Time wasted is somewhere else: if we don't have a proper design discussion, maybe something goes wrong and we have to do the work again later.*

So it would probably be necessary at times to dedicate more attention to these specific chats or to go into more depth with a voice call.

3) **The Awareness Dimension**

The awareness dimension is often part of the coordination chats. While people coordinate on their work, they also communicate with team members that they are working on some artifacts. Examples of a coordination chat with a strong awareness dimension are the coordination chats (3) and (4). Here, the coordination of the cooperation serves also to inform one the completion of a task the commencement of the next one. Whenever implicit mechanisms of awareness are insufficient, explicit **awareness chats** state what people are doing or when they will finish their work. In order to plan future work, managers ask developers what they are working on and request estimation of the time required to complete the work. Here are two examples:

[14:17:47] DK: Hi IND, looks like you are getting close to finishing your current tasks?
[14:18:52] DK: hmmm, and 559 is really not ready for execution as far as I can see....
[14:19:53] IND: Hi DK, not exactly... ya you r right.... i m finishing my tasks asap but IN704 that is on hold and i m changing it to ln progress that is really hard to complete....
[14:20:16] IND: i mean its a huge defect... so it will take lot of time to complete i think [...]
[14:21:28] DK: ok. What is your estimate?
[14:21:44] IND: i think it will take at least 2 more days to complete
[14:22:15] DK: that's ok, any technical issues, or is it just hard work?
[14:25:03] DK: yes, just saw that. Happy coding!
[14:25:43] IND: thanks...
[14:25:44] IND: :)

[05:21:45] DK: IND, are you looking at the unit tests by chance?
[05:23:34] IND: yes
[05:23:37] IND: :
[05:23:44] DK: ok, carry on :

4) **The socialization dimension**

Socialization chats are useful for team building in increasing satisfaction and motivation of team members. We report three examples. Example 13 is an introductory socializing talk for establishing the collaboration that follows.

[11:45:01] IND: hows monday treating you?
[11:45:25] IND: oh we r on the same boat
[11:45:41] DK: good to hear :

Example 14 shows the good relationship that pairs have in the collaboration. After the technical discussion, the manager gives encouraging feedback to the developer and show that he appreciates the work that has been done.

[14:24:35] DK: uh - you are very methodic. That is great
[14:24:44] IND: Thanks
[14:24:55] IND: but if i have missed any..then please let me know
[14:26:31] DK: I will - but please home now :)
In Example 15 concerns a traffic enquiry, but the ulterior motive here is to understand what the working conditions of developers are during rush times in Delhi.

[08:43:05] DK: How is traffic?
[08:48:50] IND: traffic is mor organised these days
[08:49:17] IND: lot of security and traffic police man are deployed on the road...
[08:49:33] DK: ok, is it ok to get back and forth?
[08:50:05] IND: yep..for me esp...
[08:50:30] IND: bit problem near CWG village
[08:50:49] IND: localities but ok for me from gurgaon
[08:51:01] DK: ok, good to hear

(15)

As mentioned in the introduction to this section, the social dimension often exists only as a sub-text. An example provides chat interaction (4) where the developer wants to be sure about what he has to do next. This chat has a socialization sub-text. The development team has the tendency to communicate with managers in order to double check on what to do, rather than just checking on the formal tool. This kind of interaction helps to keep a good relationship between co-workers. Also the more formal negotiation of calls in the morning seems to serve as an extended ‘Good Morning!’ Discussing the researcher’s suggestion of using tool-activated feedback to let people know of the status of a Spira™ entry, one of the project managers mentioned:

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.

I. DISCUSSION

In this section, we discuss the characteristics of the usage of different communication channels used by the virtual teams. We highlight the central role that IM plays: the glue between the heterogeneous channels. Finally, we highlight the importance of real time chats from a team building perspective not only from a coordination perspective.

A. Ecology of channels

The many communication channels exist not only side by side; rather they form an ecology of channels, a socio technical communication system where different channels are used in a complementary way. The team uses a whole range of channels. Some are structured: the source code repository and Spira™ system provide possibilities to change the state of the common artifact and provide limited possibilities to communicate through annotations related to the source code units and tasks. Other channels are unstructured, such as voice call, IM and mail. These channels can be used more freely for multiple purposes. Communication in structured channels is often formal, which means that it is defined and constrained by rules of how the work has to be performed; in unstructured channels, both formal and informal communication can take place, e.g., mails can be used for formal notifications but also for jokes or social chatting. Such a system has thus far been described with respect to an individual’s work practice [34]; below, we see how a team combines different channels.

In the CSCW discourse, it has long been known that structured channels alone are insufficient. They need to be complemented by the possibility for unstructured communication in order to negotiate how to use the structured channel [23]. The literature on global software engineering emphasizes the necessity of informal communication on unstructured channels in order to build trust and allow for informal participation [15]. In the recent research and tool support, one can observe a growing interest in the convergence of structure and structured channels; for example, the IBM Rational Team Concert [5] integrates a single application IM with structured channels, such as bug tracking, requirements definition, and versioning systems. Although technical integration is important – and our analysis indicates that the more adequate screen sharing possibility is not used as it is less integrated with the central communication environment than the Skype™ screen sharing possibility – the technology is only one side of the system we see at play.

The analysis shows that the project team has developed a common practice based on a set of social protocols [25] of how to use the different channels: Recurring patterns of coordination can be observed in our empirical material. A social protocol can be explicitly defined or it can develop implicitly with the evolution of the project. For example we observed that if a project member is present on Skype™, he is expected to answer a Skype™ request within a certain time frame. An explicit rule can be seen in Figure 1: in the DHI team, the reference number of Spira™ artifacts is used to relate to the same object through different communication channels.

Such social protocols [25] define for what purposes the different channels are used, how they are combined and how communication through different channels is related (incident number in our case). In [7], communication breakdown between sub-teams using the same set of tools is related to the use of different social protocols. Our analysis supports the importance of using the same tool and work not only with the same development environment when developing software in a distributed way, but also developing common social protocols through what sociology calls articulation work and meta-work [30]. Beyond this set of research, our analysis provides an indication that IM can play a special role in such sociotechnical communication systems.

B. IM as a glue between heterogeneous channels

Virtual teams combine different channels according to their communication needs. During the overlapping working hours, IM is used for negotiating the usage of other channels and raising activity on other channels to the awareness of team members, e.g., check in of source code or changing status of tasks in the Spira™ system. For chats used to negotiate phone calls, this phenomenon has already been investigated by Nardi [20] in 2000. In our virtual team, IM plays a crucial role not only as a way to negotiate calls, but
to coordinate all other communication channels. Even when team members are talking and sharing their screens, they exchange lines of code, links or incident numbers through the IM. We see four characteristics of IM contributing to this glue function:

- As an informal channel, IM provides the possibility of talking about code, requirements, tests, and other artifacts at the same time. In IM chats, people clarify what is formally stated in other channels, complementing the structured information in a mail or in an issue tracker system. A hypothesis for future work would be that IM provides a possibility to mediate and facilitate the work with and on assemblies of work objects and boundary objects as described in [4] [19]. For dedicated and more complex discussions, the partners may decide to switch from text-based chat to a voice call or even e-mail.
- IM provides a less obtrusive way to initiate a contact than a phone or Skype™ call. It gives the accessed partner the possibility of finishing a small task before answering.
- IM provides awareness information both with respect to the availability of team members and in lieu of a notification system. Despite the fact that Spira™ keeps track of formal assignments, thus providing a mail-based notification system, this functionality is not used. Team members often notify each other of the assignment of a task through IM.
- The nearly synchronous communication through IM seems to also provide a means for social communication for building trust and social relationships with co-workers. This is further discussed in the next section.

C. The importance of the social dimension

Only a few of the Skype™ chats are purely social, as e.g., typical chats around the water cooler in co-located settings. This might be due to Skype™ being used at the physical desk where work takes place rather than in a place away from working tools and tasks. The social dimension plays a crucial role in the team, and chat supports this dimension. Almost every chat that we analyzed had a social dimension as subtext of the main topic of discussion. Through IM, PAMs encourage developers, give positive feedbacks on their work, and joke about each other’s cultures. According to the team members, the possibility of announcing achievements (hanging over tasks) and supporting one another through small remarks and “smileys” is one of the reasons of not using the notification functionality of the Spira™ system. In our empirical material, the social dimension is tightly interwoven with the actual implementation of the work, rather than being an addendum in addition to the work; this confirms what is stated by Hinds and Kiesler in [16].

Research on the social bonding in distributed work emphasizes the importance of seeing each other [11] and on simulating co-workers presence through advanced videoconferencing systems [18]. In our case, however, video does not seem crucial from a social perspective. The relationship between team members and the sociality of the team is kept more through personal or amusing chats rather than through a video link into each other’s workplace. For team members, Skype™ seems to work as an effective substitute for face-to-face communication.

II. Conclusions

This paper has presented an ethnographic study on how different communication channels are used by a software development team distributed mainly between India and Denmark. We highlighted social protocols for the usage and the combination of different channels. Due to overlapping working hours, real time collaboration is possible across sites. IM can be seen as a type of glue between different communication and collaboration channels, and it contributes to socializing with co-workers. We identified four dimensions in chat logs: coordination, collaboration, awareness, and socialization. Although only one dimension is usually predominant, we observed that social dimension is often present as a sub-text. These social dimensions support the collaboration and help to keep a good relationship between team members.

This work is a qualitative study on only one small team, and thus considerations cannot be generalized for every software distributed team. Moreover, the analysis focuses on only the Danish/Indian side, and does not consider the manager located in Portland, USA. The collaboration with this manager was more challenging, both for Indians and for Danes, since they could not share any working hours. In fact, the manager in Portland was forced to use a different set of tools or to work during non-working hours in order to collaborate with the rest of the team. Chat in this case was not used as glue for collaboration; rather the more crucial role was played by emails.

There is a range of implications for future research. More empirical studies would help in understanding how the interlacing of different channels supports cross-site cooperation rather than focusing on individual tools. Other kinds of social software, such as microblogging, social networks and content sharing sites, could be explored. In our case IM seems to have become the space in which the collaboration around assemblies of work objects and boundary objects is mediated and facilitated. Re-analyzing the field material would contribute to a better understanding of how practitioners combine heterogeneous artefacts and communication channels to make GSD work. This, in turn, might be used to design better tools. This article only touches on the concepts of coordination mechanism and social protocols as a theoretical base for understanding the communication ecologies of distributed teams.

Acknowledgment

The authors would like to thank DHI Water and Environment for providing endorsement for this work. This research project became possible because of the collaboration of the WB project team. The authors wish to
express their gratitude to the team whose members were abundantly helpful and welcoming during the fieldwork.

REFERENCES