An Industrial Evaluation of Technological Support for Overhearing Conversations in Global Software Engineering

Kevin Dullemond, Ben van Gameren
Delft University of Technology
IHomer
The Netherlands
{*k.dullemond, b.j.a.vangameren*}*@tudelft.nl*

***Abstract*—Software engineering is by nature a highly collaborative activity. However, collaborating effectively in Global Software Engineering, in which team members are geographically, temporally and/or socio-culturally separated from each other, is more difficult. In a traditional co-located setting, one of the most important communication patterns is a (face-to-face) conversation. Technological solutions to have conversations in a distributed setting are commonly used, however overhearing conversations of others is not explicitly supported. In this paper we report on the evaluation of supporting overhearing conversations with technology in a distributed industrial setting. To do this we deployed a tool we developed with which it is possible to overhear Instant Messaging conversations in an international software development company. Based on this evaluation we report lessons learned and conclude with the most important findings of this study.**

***Keywords*-CSCW; Knowledge-based software engineering; Empirical software engineering; Tools and environments**

I. Introduction

There are strong indications that the ability to overhear conversations is valuable in carrying out your work as a Software Engineer [1]. In distributed settings however, overhearing conversations is infeasible without technological support and thus far there is no empirical evidence that it can be supported by technology in a satisfying way. In fact, there are no documented case studies researching the overhearing of conversations in such a setting at all. Therefore, to gather such empirical data, we performed a case study during a period of four months at an international software development company. The goal of this study is:

To measure the value of overhearing conversations in Global Software Engineering from actual industrial experience.

This paper is structured as follows. In the next section we discuss the background and related work of this research. In section III we discuss the research site and methods of data collection and analysis. Subsequently we present our findings in section IV and reflect upon the results and discuss limitations in section V. Finally, we conclude upon our work and discuss future research in section VI.

II. Background and Related Work

A. Lack of Awareness in GSE

More and more collaborative software engineering is no longer conducted in a single office building but in multiple dislocated office buildings or even from home. This is caused by the increasing globalization of business [2], [3], [4] and the rising popularity of working from home [5]. In collaborative work having access to the knowledge about the context in which you are working (commonly referred to as *'awareness'* [6], [7]) is essential to properly cooperate with others [6], [8]. However, when collaborating physically separated from each other team members can no longer exchange information without technological support because they do not share a common work environment. So, technological support is required to be able to gather and maintain sufficient awareness to be able to collaborate. However, the information exchange realized with technological support (e.g. telephone or e-mail) is generally inferior to information exchange in a traditional co-located office setting because in comparison it (i) takes more effort since the communication is more intentional [9], (ii) is more obtrusive [10], (iii) happens less frequently [11], [12] and (iv) contains less information [9], [13]. Therefore it is important to research and develop technological support for sharing awareness in a fashion that is as unobtrusive and effortless as possible while still providing rich and recent information.

B. Open Conversation Space

Conversations are one of the most important communication patterns that occur in a traditional office setting [14]. Advantages are [15]: integrating collaborative activities [16], [17], [18], sharing knowledge [19], [20], and creating new knowledge [21], [19], [20]. In [15] we define a conversation in the context of Global Software Engineering as: *"An exchange of information between two or more people where those participating use synchronous communication directed at the other participants"*. When working in a distributed setting having conversations is supported by IM-tools, audio conferencing and video conferencing.

However, it is not only important to have conversations yourself, but it is also important to overhear the conversations of others [15]. An example of a situation in which it is possible to do so is when working in a traditional co-located office setting. In such a setting members of the project team work physically together and converse by talking to each other. As a result, these conversations are audible by other colleagues in the office. To refer to such settings we defined the concept of an *Open Conversation Space* in [15] as: *"A space in which (i) conversations are possible between the actors in that space and (ii) these conversations are visible to other actors in that space"*.

Being able to collaborate in such an *Open Conversation Space* offers several benefits [15]:

- It provides access to the information discussed in the conversations.
- It offers the possibility of joining the conversations.
- It provides insight in the communication structure of the project team.

Collaborating in an *Open Conversation Space* can also introduce some disadvantages, for example: distractions, interruptions and a lack of privacy. However, working in a space in which the members of the project team are frequently able to both see and hear each other is so advantageous that it is one of the main reasons for working in a co-located setting [22].

C. Communico

We have discussed both the lack of awareness in GSE and the importance of an Open Conversation Space for gathering and maintaining awareness in collaborative software engineering. So, the creation of an Open Conversation Space applicable in a distributed setting (a Virtual Open Conversation Space [15]) could target the lack of awareness encountered in that setting. To this end we have developed such a space called Communico which makes it possible to overhear Instant Messaging conversations. We first reported on a prototype of Communico in [15] and on the current version of Communico in [23]. We also defined five requirements of an Open Conversation Space [23]:

REQ1. Facilitate starting conversations
REQ2. Facilitate detecting active conversations
REQ3. Facilitate monitoring active conversations
REQ4. Facilitate participating in conversations
REQ5. Facilitate the finishing of conversations

We will use these five requirements here to briefly introduce Communico in analogy to [23]. In designing Communico we focused on mimicking the traditional office setting as much as possible, because co-located software engineers enjoy awareness information benefits [24]. To implement the first requirement, we have chosen to support initiating a conversation by starting to talk to specific people. In Communico this is implemented by clicking specific people in a contact list. The second requirement, facilitating the detection of active conversations, is implemented by providing both manual and automatic support. We do this because both occur in the traditional office setting as well: people both overhear conversations by actively looking around and by being triggered by a certain event while carrying out another task. We have implemented the former by providing an active conversations list and the latter by configurable desktop alerts. The third requirement concerns monitoring active conversations. To implement this we have chosen to make conversations explicit. We did this because using implicit conversations, like in IRC, limits the creation of a structured and logical layout for group discussions [25]. We have implemented participating in conversations by allowing users to contribute, join, invite, leave and send files. Finally, we also implemented the finishing of conversations. Because a conversation requires two or more people to communicate, in Communico we chose to define the end of a conversation as the moment the total number of participants in the conversation becomes one or zero. So, when two people participate in a conversation and one of them leaves the conversation finishes. When a conversation finishes it becomes immutable: people can no longer join and as a result no content can be added to the conversation as well. In figure 1 we show a scenario of overhearing and joining a conversation with Communico.

III. Research Site and Method

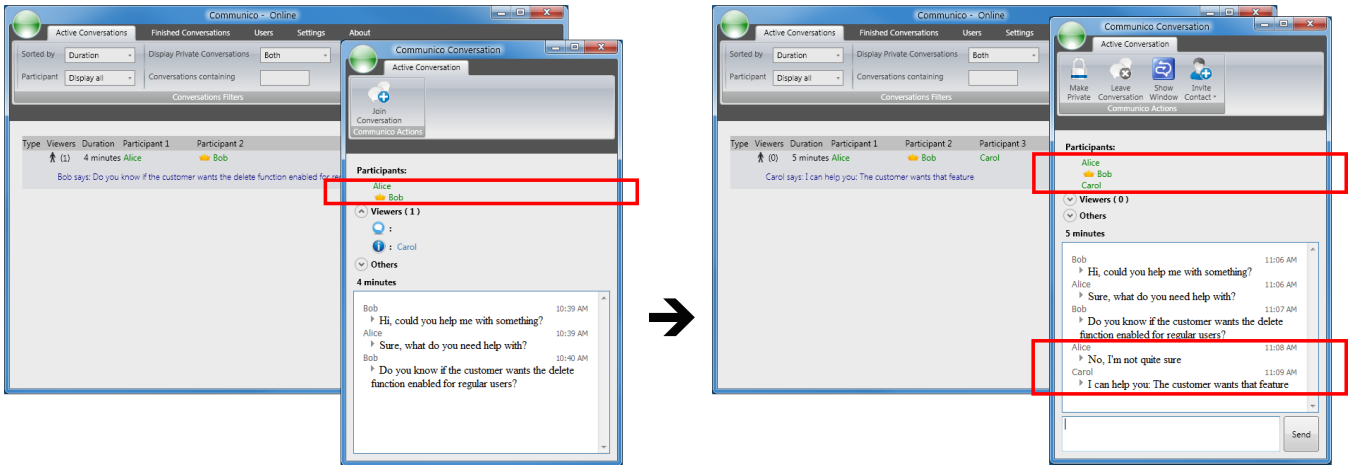
In the introduction we defined the research goal of this study: *To measure the value of overhearing conversations in Global Software Engineering from actual industrial experience.* To reach this goal we use Communico to enable the overhearing of conversations in an industrial case setting. A comparison of Communico with other existing Virtual Open Conversation Spaces can be found in table I in [23]. We choose to use Communico because we found that its conceptual choices in implementing the five requirements best mimics the traditional office setting. We investigated the following four research questions:

- How well are the benefits and challenges of having insight in active conversations exploited and alleviated?
- How well are the conversations represented?
- How well are actions to be carried out on a conversation supported?
- How well are the benefits and challenges of having insight in finished conversations exploited and alleviated?

In the remainder of this section we describe the industrial case setting and the methods with which we have investigated the research questions.

A. Site

Participants in the study are a group of Software Engineers at Exact, a Software development company operating in 40



(a) Overhearing a Conversation

(b) Joining a Conversation

Figure 1: Communico Scenario: Carol overhears the conversation Bob and Alice are having and joins the conversation

countries. Exact offers Enterprise Resource Planning software for medium-sized and small businesses. The specific group of employees that are involved in the study consists of 47 people who work on a product called Exact Online. The majority of the people in this group (42) work from the Delft office in The Netherlands, but all work from home often. Next to this, also three people participated from the Wemmel office in Belgium and two from the Minneapolis office in the USA. The majority of the people that participated in the study use Instant Messaging software on a daily basis, even when working from the same office. During the case study, which lasted four months, they also used Communico to be able to overhear conversations of the rest of the group.

B. Data Collection and Analysis Methods

To reach our research objectives we used four methods to acquire the empirical data in this study: a focus group, a semi-structured interview, a questionnaire and transactional log analysis.

1) Focus Group: We performed a Focus Group [26] to gather insights, ideas, viewpoints and opinions of people who frequently used Communico in a practical case setting. One of the main advantages of such a group setting is that it enables the participants to build on the responses and ideas of others, which increases the richness of the information gained [27].

The focus group we performed lasted approximately 2.5 hours and we selected eight frequent users of Communico from the Delft office to participate in it. Selecting participants based on their individual characteristics like this is known as purposive-sampling. We chose these people because they are more likely to have thought about the subject we wished to discuss and because they are motivated to contribute. In carrying out the focus group we followed a structured approach (see Appendix A in [28]) to ensure we would discuss the topics on which we wanted to elicit opinions. The focus group itself was conducted in a separate closed office to protect the focus group from outside influences.

2) Semi-Structured Interview: We performed semi-structured interviews [29] to gather insights, ideas, viewpoints and opinions of the interviewees. We performed two semi-structured interviews, one for the people from the Belgium office and one for the people from the US office because they could not attend the focus group. Their input is particularly valuable because the main goal of this research is to investigate the value of overhearing conversations in a distributed setting and these people worked most distributed from their colleagues. In these interviews we used the same structured approach as in the focus group.

3) Questionnaire: We chose to use a questionnaire [30] because this method makes it feasible to include the opinions of a relatively large group of people by using a standardized set of questions. In the questionnaire (see Appendix B in [28]) we asked the respondents to rate their experience in the case study on a 5-point Likert scale [31]. We included a *'no-opinion'* option to prevent people with no opinion on a specific question to answer it anyway and 'pollute' the data in this fashion [32].

We applied the Fisher's Least Significant Difference (LSD) method [33] on the ratings in each of the researched categories to reflect on their mutual importance. This method first applies the non-parametric Friedman test[1] in order to determine if the items of the data set of a specific category are significantly different. If the result of applying the Friedman test indicates this is the case, we apply the non-parametric Wilcoxon matched-pairs signed-rank test[2] to

[1]http://faculty.vassar.edu/lowry/ch15a.html
[2]http://faculty.vassar.edu/lowry/ch12a.html

pairwise compare all items in that category. From the results of this test it can be concluded whether or not it is likely one of the variables is rated as more significant.

We sent the questionnaire to 47 members of the Exact Online department of which 44 returned the survey (94% response rate). In the survey we asked people whether they frequently used Communico and only allowed users that indicated they did fill out the remaining questions. In total this concerned 25 people. The results of the questionnaire can be found in anonymized form in Appendix C in [28].

4) Transactional Log Analysis: Transaction Log analysis is a data collection method for analyzing system performance and user behavior [34]. The main benefits of using transaction log analysis to analyze user behavior are that it is an unobtrusive method and gathers much more data than any data set obtained via surveys, laboratory studies or by user observation in naturalistic settings [34]. We use the method in this study to gather data on usage frequency of the conversation overhearing functionality to be able to reflect on the adoption rate of our solution.

IV. Findings

We present the findings of the empirical study in five parts. Firstly, we reflect on the four research questions. We do this by presenting the benefits and challenges of an *Open Conversation Space* and what information and actions are useful in such a setting. These characteristics were identified in an empirical case study about the value of the conceptual idea of overhearing conversations based on the working experience of a group of software engineers [1]. Subsequently, we check for each of these characteristics whether we can draw statistically valid conclusions regarding the relative ordering of how well they are implemented. If this is the case we show their relative ordering by presenting a table summarizing all comparisons. In this table a (green) *'larger than'*-sign means the item on the left is rated as more important to a statistically significant level. We show a (red) *'X'* when we cannot draw statistically significant conclusions regarding the relative importance of two items. Finally, we analyze the user behavior: how the users of Communico interacted with the system. For each of the five parts, we discuss the findings, discuss possible improvements of Communico and present lessons learned. For each of the characteristics, a detailed presentation can be found in Appendix D in [28] which includes all of the data we gathered, a descriptive analysis of this data illustrating its distribution and a complete analysis to determine the relative ordering.

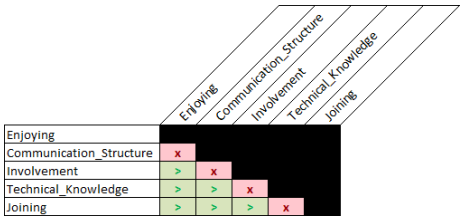
A. Benefits and challenges of overhearing conversations

1) Benefits: The benefits of having insight in the active conversations we identified in [1] are the following:

- Having access to the technical knowledge of colleagues — `Technical Knowledge`
- Acquiring involvement with colleagues — `Involvement`
- Enjoying your work — `Enjoying`
- Being able to join a conversation — `Joining`
- Acquiring insight in the communication structure of the team — `Communication Structure`

When applying the Friedman test on the gathered data it showed the variables are likely to come from a different distribution ($\chi^2(4) = 29.651, P = 0.000$). So, we applied the Wilcoxon test on all pairs of 2 variables in the data set to check if we could conclude anything about their mutual importance. The results of these tests are summarized in table I.

Table I: Comparative Analysis - How well Communico exploits the benefits of overhearing conversations



From table I we may conclude that *Joining* is exploited more than *Involvement* (Z=-2.725, P=0.006), *Communication_Structure* (Z=-3.397, P=0.001) and *Enjoying* (Z=-3.502, P=0.000). Next to this, we can conclude that *Technical_Knowledge* is exploited more than *Communication_Structure* (Z=-2.517, P=0.012) and *Enjoying* (Z=-3.147, P=0.002). Finally, we can also conclude that *Involvement* is exploited more than *Enjoying* (Z=-2.696, P=0.007).

In the focus group the benefits reported to be exploited best by Communico were *Joining* and *Technical_Knowledge*. In the interviews these two were also seen as well exploited by Communico, however the interviewees also reported a significant increase in *Involvement* with the rest of the team due to being able to overhear their conversations with Communico. One of the interviewees from Belgium said: *"I felt more like being there"* while one of the interviewees from the USA said: *"In the morning I would scroll through all the conversations my colleagues had during their working day that far. This made me feel more connected to them"*. It is likely the interviewees felt different than the focus group participants because the people that were interviewed worked dislocated from nearly all of their colleagues while the distribution was significantly less for the participants of the focus group.

Lesson Learned 1
Overhearing conversations with technological support results in a stronger increase of involvement with colleagues for people that work more dislocated from their colleagues than for those that work more co-located

Finally, participants of the focus group reported the *Communication_Structure* as not being exploited particularly well. They suggested making the relations explicit in the form of a graph, since manually checking who are communicating often takes too much effort. It is noteworthy that with respect to the benefits only a possible improvement was mentioned in relation to the exploitation of the *Communication Structure* while in [1] this is rated as one of the least important benefits.

Lesson Learned 2
Only showing the conversations is not sufficient to acquire insight in the communication structure of the team

2) Challenges: The challenges of having insight in the active conversations we identified in [1] are the following:

- It can be distracting from the current work activities `Distracting`
- The context of the conversation can be unclear `Context`
- The information is volatile `Volatile`
- A lack of control for the people whose conversations are overheard `Lack Of Control`

In the focus group and interviews it was discussed that *Volatile* is not a large problem since this is something inherently tackled by a tool such as Communico which saves conversations. The other three challenges of overhearing conversations are however also encountered when using Communico and in [1] the two challenges rated as most important are: *Distracting* and *Context*. The focus group identified several improvements of Communico to help alleviate these. With respect to *Distracting* the participants of the focus group suggested making it possible to withdraw yourself from the Open Conversation Space completely by putting on *'virtual head phones'*. Subsequently, when you take off these *'virtual head phones'* it should be possible to get some sort of summary of the conversations that occurred while you were occupied. Another method proposed was to limit the amount of conversations you overhear by creating *'virtual office walls'* separating yourself from certain groups of people while still overhearing the conversations of others.

Lesson Learned 3
Having control over the amount of conversations you overhear is important to limit distractions

Further, the focus group discussed that the relations between conversations are an important part of the context of a conversation and suggested to make it possible to either automatically or manually link related conversations to each other.

Lesson Learned 4
Improving having insight in the relations between conversations also improves the clarity of the context

Finally, the participants argued there was insufficient control with respect to making conversations private. Firstly, they argued they wanted to be able to make conversations private before they start and secondly that they would like to see private conversations removed from the active conversations list altogether. Currently, private conversations are shown there but their content is hidden from non-participants. Participants argue they see no value in knowing others are having a private conversation and they do not want others to know of their own private conversations as well.

Lesson Learned 5
Only enabling making conversations private after initialization provides insufficient control for the people whose conversations are overheard

Lesson Learned 6
Making the fact that private conversation are occurring visible to non-participants provides insufficient control for participants in private conversations

B. Information about a conversation

The important types of information about a conversation we identified in [1] are the following:

- Who are Participating in the conversation `Participating`
- Who are viewing the conversation `Viewers`
- The complete factual content `Content`
- The commitment of a participant `Commitment`
- The contribution of a participant `Contribution`
- The subject of the conversation `Subject`
- The tone of the conversation `Tone`
- The type of the conversation `Type`
- The phase the conversation is in `Phase`
- The location the conversation takes place `Location`
- The accessibility of the conversation `Accessibility`

The focus group and interviews elicited similar opinions about the importance of the information items. Both considered it important to be able to tag conversations with the *Subject* to be able to quickly and unobtrusively decide whether it is interesting. In [1] the *Subject* was also elicited as the most important information item to accomplish this

Table II: Comparative Analysis - How well Communico shows the information items about a conversation

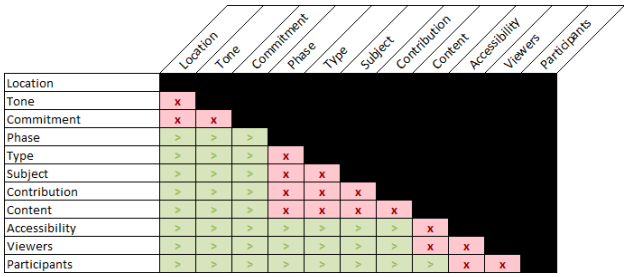
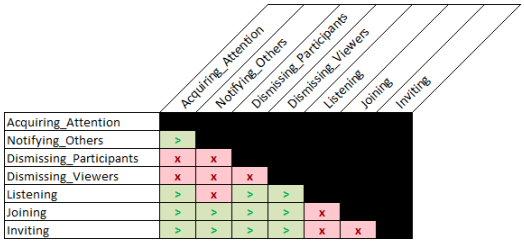


Table III: Comparative Analysis - How well Communico supports the actions to be carried out on a conversation



and in the current version of Communico it was rated relatively low (see table II). One of the participants said: *"I used the join feature less than would have been possible if the automatic detection of conversations had been better"*.

Lesson Learned 7
Deciding whether a conversation is interesting can be done quicker and less obtrusive when its subject is known

Next to this, showing the last two sentences instead of the last one can also improve conversation detection since the last sentence is often an acknowledgment like: *'sure'*, *'alright'* or *'I'll get right on that'*. As a final way to improve conversation detection, some participants proposed to include a text based sliding text ticker to find out about a conversation in Communico. The advantage of such a ticker is that it is less disruptive than a desktop alert while being easier to continually scan than the active conversations list.

C. Actions possible on a conversation

The actions that are possible with respect to a conversation we identified in [1] are the following:

- Joining a conversation — `Joining`
- Inviting someone to join a conversation — `Inviting`
- Listening to a conversation — `Listening`
- Dismissing other participants — `Dismissing Participants`
- Dismissing viewers — `Dismissing Viewers`
- Acquiring the attention of the participants — `Acquiring Attention`
- Notifying others of the conversation — `Notifying Others`

In the focus group it was discussed that while *Joining* is supported adequately (see table III) it would be preferable for the join-process to be more like the co-located setting. Currently, someone has to request to join the conversation and explicit permission needs to be given for this to be allowed. The participants of the focus group suggested changing this so that people automatically join a conversation when they start to talk in a conversation they are watching. One of the participants in the focus group said: *"Since most people accept a join request anyway it is best to allow joining by default and make it possible to dismiss a person that joined later if this is undesirable"*.

Lesson Learned 8
An implicit join process is preferable to an explicit join process

Additionally, by making it possible to explicitly dismiss participants the original participants of the conversation still have the option to ask the newly joined participant to leave. In [1] *Joining* and *Inviting* are rated as most important so even though their realization is rated best here as well (see table III) it makes sense that focus group participants propose improvements to the joining process. It is also worth mentioning that in [1] we concluded adding participants to a conversation is more important than removing people. So, the fact that focus group participants specifically proposed to add an explicit dismiss participant option indicates that removing participants is in fact important, even if it is less important than adding participants to a conversation. Finally, the focus group participants also suggested to support explicitly notifying people outside the conversation of a conversation that might be interesting to them and to include something like an attention buzzer to acquire the attention of the other participants in the conversation.

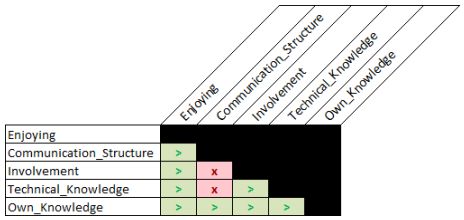
D. Benefits and challenges of finished conversations

1) Benefits: The benefits of having access to finished conversations we identified in [1] are the following:

- Having access to knowledge you might otherwise forget — `Own Knowledge`
- Access to technical knowledge of colleagues — `Technical Knowledge`
- Acquiring involvement with your colleagues — `Involvement`
- Enjoying your work — `Enjoying`
- Acquiring insight in the communication structure — `Communication Structure`

Having access to your own knowledge was rated as exploited best (see table IV). This viewpoint was shared

Table IV: Comparative Analysis - How well Communico exploits the benefits of having access to finished conversations



by the participants of the focus group and interviewees. However, like for the benefits of overhearing active conversations, the (dislocated) interviewees voiced the opinion that *Involvement* is also exploited particularly well.

Lesson Learned 9
Having access to finished conversations results in a stronger increase of involvement with colleagues for people that work more dislocated from their colleagues than for those that work more co-located

The improvements of Communico identified with respect to these benefits were mostly similar to those identified for the benefits of overhearing active conversations. However, because the list of finished conversations is considerable longer than the list of active conversations the focus group suggested the addition of a date range filter to make this list more manageable. In [1] we also suggested it is particularly important to find ways to extract data from the set of finished conversations because having access to your *Own Knowledge* and to *Technical Knowledge* were the two benefits of having access to finished conversations that were found to be most important.

2) Challenges: The challenges of having access to the finished conversations we identified in [1] are the following:

- It can be distracting from the current work activities `Distracting`
- The context of the conversation can be unclear `Context`
- A lack of control for the people whose conversations are overheard `Lack Of Control`

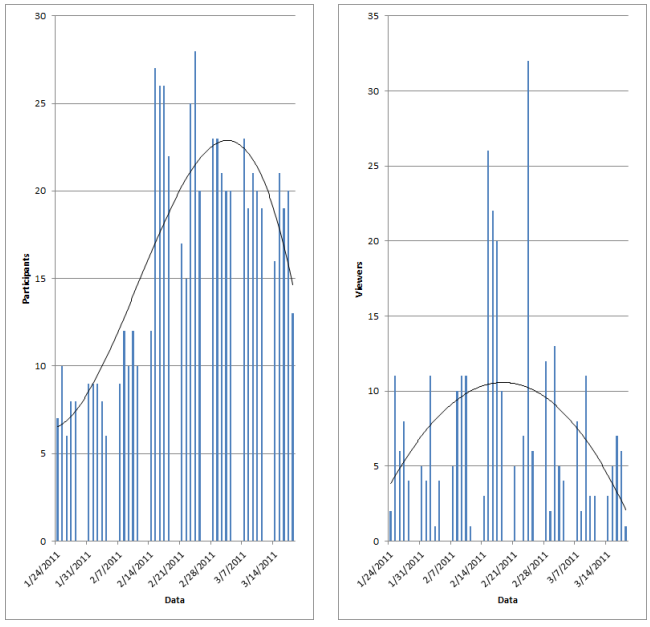
In the focus group and interviews the same limitations and possible improvements of Communico were discussed as for the challenges of overhearing active conversations with the addition that participants suggested alleviating *Lack Of Control* by making it possible to make conversations private after they finish because people often forgot to do this.

Lesson Learned 10
Only enabling making conversations private before they finish provides insufficient control for the participants of the conversation

E. Behavior analysis

To be able to analyze how the users of Communico interacted with the system we used automatically generated transaction logs. From these logs we derived that in the total four months of usage 53 unique users used Communico for 4185 hours in total. The average length of a user session was about 4.4 hours. The users had 1921 conversations in total. The average number of participants of a conversation was 2.2 and the highest number of participants in a conversation was 9. During this period 605 view actions took place at 493 different conversations, the average number of viewers of a conversation was 0.31 and the highest number of viewers was 5.

We divided the total period of four months of usage of Communico in two parts of two months. In the first two months we deployed Communico to a select number of people to test the usage of Communico in the specific setting of Exact, resolve problems and adapt Communico to best fit the needs of this specific setting. In the subsequent two month period we made Communico available to the entire department on a voluntary basis. In the period Communico was available to the entire department we identified a trend in the use of the tool. After we made the tool available, the usage increased, peaked and subsequently decreased when we stopped actively promoting its use. This can be seen in figure 2 where we show the increase and subsequent decrease in total number of participants and total number of view actions per day for all users of Communico. Especially



(a) Number of users

(b) Total number of view actions per day

Figure 2: Behavior analysis

in the second graph we see a strong decrease in the number of view actions while in the other graph we also see a decrease of use near the end following a peak.

Lesson Learned 11
In settings where part of the team works co-located the use of technology, supporting the overhearing of conversations without offering specific advantages over the co-located setting, will strongly decrease over time.

It is also interesting to see the relatively large reduction in number of view actions relative to the reduction in participants.

Lesson Learned 12
A decrease in the number of participants of a tool supporting the overhearing of conversations will result in a stronger decrease in the number of overhear actions

V. Discussion

In this section we will reflect on the findings and discuss the most important results. Before this discussion it is important to re-emphasize the context in which this discussion takes place: a practical case setting in which we supported the overhearing of conversations. Outside of this scope several of the lessons learned we have introduced in the previous section and discuss in this section, have already been published. For example in relation to Lesson Learned 2, Sarma et al. [35] report on their tool Tesseract which shows the social network of developers as determined by their communication records, which in their case concerned email communication, comments about a bug and work performed and submitted in the bug tracker. In the rest of the discussion we focus on the value of overhearing conversations in Global Software Engineering.

To start, we found that, when deploying awareness sharing technology into settings where part of the team works co-located, the use of this technology should offer specific advantages over the co-located setting to stimulate the use by people that work mostly co-located. This stimulation is required because of a combination of three factors. Firstly, we found in the use of Communico that on the one hand, the people working mostly distributed value the overhearing of conversations with Communico most since it gives them access to awareness information they did not have access to before. On the other hand however, the people working mostly co-located value the overhearing of conversations less because these people already overhear a significant portion of the conversations outside of Communico because they can communicate face-to-face (Lesson Learned 1). Therefore the mostly co-located people have less incentive to use the tool. Secondly however, it is necessary the mostly co-located people also use Communico to actually make the technological support for overhearing conversations work. Since without their cooperation, their conversations are still inaccessible to and cannot be overheard by, the distributed people. Finally, we also identified a strong decrease in the use of Communico at the end of the study, when we were no longer present daily, indicating that our presence artificially stimulated the use.

Related to the previous finding we also found that to stimulate acceptance, and therefore use, of awareness sharing technology, it is important to provide the users with more control over the information about them which is being shared (this includes their actions). This can also be seen in some of the lessons learned discussed in the previous section. Firstly, people propose to enable making conversations private before a conversation starts (Lesson learned 5) and after it finishes (Lesson Learned 10). Next to this, people also propose to not show private conversations in the active conversations list to non-participants (Lessons Learned 6).

Another important result we identified is that being able to properly detect interesting conversations is essential. In a traditional office setting this happens unobtrusively and it is important to approach this standard when constructing technological support to enable the overhearing of conversations. This can also be seen in the ideas for future improvements identified is this paper. Examples of this are being able to automatically identify the subject of and relations between conversations (Lessons Learned 7 and 4). Related to this, also the prevention of information overload is important. When we are better capable of detecting when a conversation is interesting, we will also be better at preventing people from getting too much information (Lesson Learned 3). An additional factor in determining how much information and what granularity of information is needed has to do with the current activity of the user. An example mentioned in the focus group is to put on *'virtual head phones'* to withdraw yourself from the open conversation space and avoid disruption when performing a task that requires significant attention.

A. Threats to Validity

A threat to external validity is that we only studied one department in a single company. To be able to generalize the findings the study should be repeated in more settings. Next to this, also the size of the sample is a threat to external validity. For practical reasons, we performed one focus group and sent the questionnaire to 47 people of which 25 reported having used Communico actively. Also the sample we selected for the focus group may not be representative as well since we selected all participants from the Delft location for practical reasons. To mitigate this risk we performed semi-structured interviews with the dislocated people.

There are also threats to internal validity. Firstly, most people worked from the Delft office on a daily basis. However, most participants were used to working in a distributed setting, often worked from home and there were

three participants from Belgium and two from the USA as well. It is possible that the limited exposure of a portion of the participants to working distributed from their colleagues caused items to be misrated in the questionnaire. Next to this, the people that participated in the focus group and semi-structured interviews also participated in the subsequent questionnaire. This could have biased the results due to a learning effect caused by repeated testing.

We attempted to mitigate threats to reliability by rigorously describing our research site and methods and making all of our quantitative data available online. We do this in an attempt to make, both our data gathering methods and the analysis of our data, repeatable. Subsequently, a threat to construct validity is mono-operation bias. Because we only researched supporting the overhearing of conversations with technology with one specific tool one could argue the results only apply to the use of that tool. We mitigated this threat by defining an explicit set of requirements of open conversations spaces and discussing how Communico fulfills these.

Finally, there is also a threat to statistical conclusion validity. In our analysis of the survey data we used Fishers' Least Significant Difference method to help reduce the number of false positives caused by the pair-wise comparison of all items. In comparison with other methods which aim to accomplish this, Fishers' LSD is fairly liberal. We chose to use a fairly liberal method because of the exploratory character of this research.

VI. Conclusions and Future Work

In this paper we have reported on an empirical study about the evaluation of supporting overhearing conversations in a distributed setting with technology. The goal of the study is: *To measure the value of overhearing conversations in Global Software Engineering from actual industrial experience*. The most important results of this paper are:

- The value of awareness sharing technology is higher for people that work more distributed from their colleagues
- The value of awareness sharing technology is higher when a larger portion of a team uses them
- In settings where part of the team works co-located the use of awareness sharing technology should offer specific advantages over the co-located setting to stimulate the use by people that work co-located
- The acceptance, and therefore the value, of awareness sharing technology can be increased by providing the users with more control over the information about them which is being shared
- The value of technological support for overhearing conversations in distributed settings is higher when such support can more accurately detect interesting conversations while preventing an overload of information

Future work will concern the investigation of how to deal with settings in which a portion of the team is distributed while another portion mainly works co-located. We feel this is not only essential to the success of a tool supporting the overhearing of conversations but for awareness sharing technology in general.

References

[1] K. Dullemond, B. van Gameren, and R. van Solingen, "An Exploratory Study on Open Conversation Spaces in Global Software Engineering," in *Proceedings of the 2011 International Conference on Collaborative Computing: Networking, Applications and Worksharing*. IEEE, 2011.

[2] E. Carmel, *Global software teams: collaborating across borders and time zones*. Upper Saddle River: Prentice Hall PTR, 1999.

[3] J. Herbsleb and D. Moitra, "Guest Editors' Introduction: Global Software Development," *IEEE Software*, vol. 18, no. 2, pp. 16–20, 2001.

[4] J. Herbsleb, "Global Software Engineering: The Future of Socio-technical Coordination," in *Proceedings of the IEEE 2007 Workshop on the Future of Software Engineering*. IEEE Computer Society Press, 2007, pp. 188–198.

[5] The Dieringer Research Group Inc., "Telework Trendlines 2009: A Survey Brief by WorldatWork," 2009.

[6] K. Schmidt, "The Problem with 'Awareness': Introductory Remarks on 'Awareness in CSCW'," *Computer Supported Cooperative Work*, vol. 11, no. 3-4, pp. 285 – 298, 2002.

[7] P. Dourish and V. Bellotti, "Awareness and Coordination in Shared Workspaces," in *Proceedings of the ACM 1992 Conference on Computer Supported Cooperative Work*. ACM Press, 1992, pp. 107–114.

[8] A. Syri, "Tailoring cooperation support through mediators," in *Proceedings of the 1997 European Conference on Computer Supported Cooperative Work*. Kluwer Academic Publishers, 1997, pp. 157–172.

[9] C. Gutwin, R. Penner, and K. Schneider, "Group awareness in distributed software development," in *Proceedings of the ACM 2004 Conference on Computer Supported Cooperative Work*. ACM Press, 2004, pp. 72–81.

[10] J. Fogarty, S. Hudson, C. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. Lee, and J. Yang, "Predicting human interruptibility with sensors," *ACM Transactions on Computer-Human Interaction*, vol. 12, no. 1, pp. 119–146, 2005.

[11] J. Herbsleb and A. Mockus, "An empirical study of speed and communication in globally distributed software development," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 481–494, 2003.

[12] T. Allen, *Managing the flow of technology*. MIT press, 1977.

[13] J. Olson and S. Teasley, "Groupware in the wild: lessons learned from a year of virtual collocation," in *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work*. ACM Press, 1996, pp. 419–427.

[14] D. Perry, N. Staudenmayer, and L. Votta, "People, organizations, and process improvement," *Software, IEEE*, vol. 11, no. 4, pp. 36 –45, Jul. 1994.

[15] K. Dullemond, B. van Gameren, and R. van Solingen, "Virtual open conversation spaces: Towards improved awareness in a GSE setting," in *Proceedings of the 2010 International Conference on Global Software Engineering*. IEEE Computer Society Press, 2010, pp. 247–256.

[16] J. Espinosa and E. Carmel, "The impact of time separation on coordination in global software teams: a conceptual foundation," *Software Process: Improvement and Practice*, vol. 8, no. 4, pp. 249–266, 2003.

[17] S. Greenberg and C. Gutwin, "A descriptive framework of workspace awareness for real-time groupware," *Computer Supported Cooperative Work*, vol. 11, no. 3-4, pp. 411–446, 2002.

[18] Y. Ren and R. Kraut, "A Simulation for Designing Online Community: Member Motivation, Contribution, and Discussion Moderation." Manuscript in preparation: Retrieved from: http://www.cs.cmu.edu/~kraut on June 2^{nd} 2011.

[19] A. Webber, "What's so new about the new economy?" *Harvard Business Review*, 1993.

[20] T. Erickson, D. Smith, W. Kellogg, M. Laff, J. Richards, and E. Bradner, "Socially translucent systems: social proxies, persistent conversation, and the design of babble," in *Proceedings of the SIGCHI 1999 Conference on Human Factors in Computing Systems*. ACM Press, 1999, pp. 72–79.

[21] E. Wynn, "Office conversation as an information medium," Ph.D. thesis, University of California, Berkely, 1979.

[22] R. Kraut, R. Fish, R. Root, and B. Chalfonte, "Informal communication in organizations: Form, function, and technology," in *Human reactions to technology: Claremont Symposium on Applied Social Psychology*. Sage Publications, 1990, pp. 145–199.

[23] K. Dullemond, B. van Gameren, and R. van Solingen, "Overhearing Conversations in Global Software Engineering - Requirements and an Implementation," in *Proceedings of the 2011 International Conference on Collaborative Computing: Networking, Applications and Worksharing*. IEEE, 2011.

[24] I. Omoronyia, J. Ferguson, M. Roper, and M. Wood, "Using developer activity data to enhance awareness during collaborative software development," *Computer Supported Cooperative Work (CSCW)*, vol. 18, pp. 509–558, 2009.

[25] G. Trapani and A. Pash, "The complete guide to Google Wave," 2008, Retrieved from: http://completewaveguide.com on February 24^{th} 2010.

[26] J. Kontio, L. Lehtola, and J. Bragge, "Using the focus group method in software engineering: Obtaining practitioner and user experiences," *Empirical Software Engineering, International Symposium on*, vol. 0, pp. 271–280, 2004.

[27] J. Langford and D. McDonaugh, *Focus Groups: Supporting Effective Product Development*. Taylor and Francis, 2003.

[28] Dullemond, K. and van Gameren, B. and van Solingen, R., "An Industrial Evaluation of Technological Support for Overhearing Conversations in Global Software Engineering," Delft University of Technology, Tech. Rep., 2011, http://www.aspic.nl/publications/TechReport001.pdf.

[29] A. Fontana and J. Frey, "The interview: From neutral stance to political involvement," *The Sage handbook of qualitative research*, vol. 3, pp. 695–727, 2005.

[30] A. Fink, *How to manage, analyze, and interpret survey data*, 2nd ed. London: Sage, 2003.

[31] R. Likert, "A technique for the measurement of attitudes." *Archives of psychology*, vol. 22, no. 140, pp. 1–55, 1932.

[32] K. D. Bailey, *Methods of Social Research*. New York: Free Press, 1978.

[33] R. Fisher, "The design of experiments." 1935.

[34] B. Jansen, I. Taksa, and A. Spink, "Research and methodological foundations of transaction log analysis," in *Handbook of research on Web log analysis*. Hershey, 2008, pp. 1–17.

[35] A. Sarma, J. Herbsleb, L. Maccherone, and P. Wagstrom, "Tesseract: Interactive visual exploration of socio-technical relationships in software development," in *Proceedings of the 2009 IEEE 31st International Conference on Software Engineering-Volume 00*. IEEE Computer Society Washington, DC, USA, 2009, p. 2333.