Categorizing communicability evaluation breakdowns in groupware applications

Raquel Oliveira Prates[1,2]
[1]Department of Computer Science
State University of Rio de Janeiro, UERJ
R. São Francisco Xavier, 524 - 6o. andar
20550-013
Maracanã, Rio de Janeiro, RJ, Brazil
raquel@ime.uerj.br

Clarisse Sieckenius de Souza[2]
[2]Department of Informatics
Catholic University of Rio de Janeiro
R. Marquês de São Vicente, 225
22453-900
Gávea, Rio de Janeiro, RJ, Brazil
clarisse@inf.puc-rio.br

Patrícia Seefelder de Assis[2,3]
[3]Instituto Politécnico, IPRJ,
State Univeristy of Rio de Janeiro, UERJ
Caixa Postal 97282, 28601-970
Nova Friburgo, RJ, Brazil
patricia@iprj.uerj.br

Abstract

The communicability evaluation method evolved within the Semiotic Engineering framework and its main goal is to assess how designers communicate to users, through the interface, both their design intent and the interactive principles they have selected for the application. The method consists of 3 steps: tagging, interpretation, semiotic profiling and was originally developed to evaluate how well users get the designer's message from interacting with single-user interfaces. In order to extend the communicability evaluation method to account for groupware applications, we must identify new utterances and problem categories that apply to interacting with and through these applications. In this paper we take the first step in that direction and, based on the results of two case studies, we propose four types of problems that should be added to the original set of HCI problems to characterize interactive breakdowns in groupware applications.

Introduction

Research and practice in Human Computer Interaction (HCI) have fostered the development and application of theories and methods to support design, development and evaluation of interfaces so as to allow people to interact with them efficiently, effectively and with satisfaction [Hartson 1998]. The best known and most frequently applied theoretical approach is Cognitive Engineering, which is based on cognitive psychology and has as its main goal to describe the stages that users go through when performing an action. The designers who take this approach aim at minimizing the gap between the users' goals and intentions, on one side, and the actions they have to perform to achieve them, on the other [Norman 1986, Norman 1988].

We take a Semiotic Engineering approach [de Souza, 1993, de Souza et al., 2000] to HCI. This approach is based on Semiotics (the discipline that studies communication and signification [Eco, 1976]) and views the interface as a one-shot message from designer to users, by which designers tell users (1) who they think the users are, (2) what range of goals users can achieve with the application, and (3) what messages the users can or should exchange with the application to achieve those goals. The Semiotic Engineering approach is complementary to the Cognitive Engineering one. Whereas the latter focuses on how to present the application in a way that is easier for the user to understand, the former focuses on the designer's communicative intent and on the design process, so that communication is efficient and effective.

In our theoretical framework we have defined the concept of communicability [de Souza et al. 1999, Prates et al 2000a, Prates et al. 2000b] as the property of software that efficiently and effectively conveys to users its underlying design intent and interactive principles. Thus, the goal of the communicability evaluation method is to let designers appreciate how well users are getting the intended messages across the interface and to identify communication breakdowns that may take place during interaction. In a design process the communicability evaluation method will add to usability evaluation ones.

Originally the communicability evaluation method was developed to assess the designer-user communication in single-user interfaces. In order to extend it to account for groupware applications, we should take into consideration not only the user-system interaction, but also user-user interaction through the system. Which aspects of groupware should be evaluated and which methods would be most efficient to do so

are issues still under investigation [Baker et al. 2001, Gutwin & Greenberg 2000, Cugini et al 1997, Blythin et al 1997]. The new problem categories we propose to the communicability evaluation method try to characterize communicative breakdowns that are specific to a message being sent from a designer to a group of people who will use computers with a variety of collective goals, as well as those being sent from user to user.

One of the challenges for groupware applications is to prevent the technology from intensifying the (potential) problems that groups typically face, such as unproductive competition, disruptive behavior, cultural differences and mutual misunderstandings. Likewise, one of the challenges for groupware evaluation methods is to distinguish between group interaction problems that are somehow due to HCI design from those that originate basically from human behavior in social contexts. In this proposal we will be addressing communicative breakdowns that we suppose are caused by HCI design choices.

In the next section we present the original communicability evaluation method. We then describe the two case studies that were carried out. It is based on the results of these case studies that we propose a first step towards extending the communicability evaluation method to groupware applications. In the following section we present the interaction levels in which breakdowns that affect the group's interaction may occur. Finally, we introduce four categories of problems in groupware applications that should be added to the set of HCI problems used in the original communicability evaluation method. At last we present our conclusions and the next steps in our research.

The Communicability Evaluation Method

To apply the communicability evaluation method, evaluators must identify tasks to be performed by users and the users' interaction with the application must be recorded using a software application that is able to capture mouse-pointer movements and other screen events. Once the interaction is recorded, the evaluator analyzes the resulting movie and performs the 3 steps of the method: tagging, interpreting and semiotic profiling.

Tagging

In order to perform the tagging the evaluator is provided with a predefined set of tags (see Table 1). Each one of these tags represent an utterance that expresses a user's reaction to what happens during interaction (e.g., “Oops!” or “What's this?” or “Where is that?” and the like) when conversational breakdowns occur. The evaluator plays the interaction movie and identifies sequences of actions that represent these breakdowns and then chooses an utterance that describes it. This step could be perceived as "putting words into the user's mouth".

Tag	Description
• **Where is?** • **What now?**	The user seems to be searching for a specific function but demonstrates difficulty in locating it. So, he sequentially (worse case) or thematically (better case) browses menus and/or toolbars for that function, without triggering any action. This category includes a special case we have called **What now?** which applies when a user is clearly searching for a clue of what to do next and not searching for a specific function that he hopes will achieve what he wants to do.
• **What’s this?** • **Object or action?**	The user seems to be exploring the possibilities of interaction to gain more (or some) understanding of what a specific function achieves. He lingers on some symbol waiting for a tool tip and/or explicitly calls for help about that symbol, or he hesitates between what he thinks are equivalent options. This category also includes cases in which users are confused about widgets being associated with objects instead of actions and vice versa (**Object or action?**).
• **Oops!** • **I can’t do it this way.** • **Where am I?**	This category accounts for cases in which a user performs some action to achieve a specific state of affairs, but the outcome is not what he expected. The user then either immediately corrects his decision (typically *via* Undo or by attempting to restore some previous state) or completes the task with an additional sequence of actions. Sometimes the user follows some path of action and then realizes that it’s not leading him where he expected. He then

	cancels the sequence of actions and chooses a different path. In this case the associated utterance is **I can't do it this way**. This category includes another one, **Where am I?**, in which the user performs some action that is appropriate in another context but not in the current one.
• **Why doesn't it?** • **What happened?**	This category involves cases in which the user expects some sort of outcome but does not achieve it. The subsequent scenario is that he then insists on the same path, as if he were so sure that some function should do what he expects that he simply cannot accept the fact that it doesn't. Movies show that users carefully step through the path again and again to check that they are not doing something wrong. The alternative scenario (**What happened?**) is when they do not get feedback from the system and are apparently unable to assign meaning to the function's outcome (halt for a moment).
• **Looks fine to me…**	The user achieves some result he believes is the expected one. At times he misinterprets feedback from the application and does not realize that the result is not the expected one.
• **I can't do it.**	The user is unable to achieve the proposed goal, either because he does not know how to or because he does not have enough resources (time, will, patience, etc.) to do it.
• **Thanks, but no, thanks.** • **I can do otherwise.**	The user ignores some preferential intended affordance present in the application's interface and finds another way around task execution to achieve his goal. If the user has successfully used the afforded strategy before and still decides to switch to a different path of action, then it is a case of **Thanks, but no, thanks**. If the user is not aware of the intended affordance or has not been able to use it effectively then it is a case of **I can do otherwise**. Whereas **Thanks, but no, thanks** is an explicit declination of some affordance, **I can do otherwise** is a case of missing some intended affordance.
• **Help!**	The user accesses the help system.

Table 1 - Set of tags used to describe interactive breakdowns in single-user applications

For instance, if the user's task is to change into bold the title of a text in an editor, and the evaluator notices in the movie that the user is looking for the bold function in the interface, he tags that interactive sequence with the tag "Where is (the bold function)?". This is the equivalent of saying that "if the user were to say anything, I believe he would be asking where he could find the bold function". Thus, this step can be perceived as an attempt to recreate the users' verbal protocol.

Interpretation

In this step the evaluator tabulates the utterances associated to the identified breakdowns and associates them onto HCI categories of problems or design guidelines. The general classes of problems identified are navigation, meaning assignment, task accomplishment, missing and declining affordances. HCI experts may use alternative taxonomies for a more precise user-system interaction diagnosis. Whereas the first three classes are well known in HCI, missing and declining affordances are not usually identified by other methods. Declining an affordance means that the user understands what the designer thought would be the best way to carry out a task, but decides otherwise. Missing an affordance indicates that the user does not understand the designer's intentions at all for the corresponding task.

Associating taggings with these problem categories can be done straightforwardly most of the time. However, some utterances may correspond to more than one category. In these cases the context must be used to clarify which problem would be the most appropriate one. Heuristically, "Where is?", "What now?", "Oops!", "I can't do it this way.", and "Where am I?", all point to navigation problems. "What's this?", "Object or action?", "Why doesn't it?", "What happened?", "Oops!", "I can't do it this way.", "Where am I?", "Looks fine to me…" can be related to meaning assignment problems. "I can't do it." and "Looks fine to me…" can be associated to task accomplishment problems.

Finally, breakdowns tagged as “Thanks, but no thanks.” are mapped onto declination of affordance, and those tagged as “I can do otherwise.” are mapped onto missing of affordance.

Semiotic Profiling

The semiotic profiling step must be performed by Semiotic Engineering experts, since its goal is to attempt to retrieve the original designer’s meta-communication, that is, the meaning of the overall designer-to-user message. In order to do that, evaluators must cast their interpretation of the resulting tabulation from the previous step in semiotic terms.

Case Studies in Groupware

We have carried out two case studies with the purpose of getting some indication of whether a new set of utterances and problem categories would be needed to characterize interactive breakdowns in groupware applications, and if so which would those be. We have taken a top-down approach and started by focusing on the problem categories. We performed two case studies using NetMeeting™ [Microsoft]. One involved a group of novices who had never used the Internet before, and the other with system developers who do not regularly use chat rooms. The novice users had to achieve tasks in two different scenarios involving only the chat function of NetMeeting. The system developers had to achieve tasks using the whiteboard and the chat functions in only one scenario.

Case Study 1 - Novices

In this case study, three users were observed. None of them was a proficient Windows user, nor had they ever entered the Internet. They were asked to perform tasks in two different scenarios using the NetMeeting chat. In the first part of the experience they had to accept an incoming call and take part in a conversation with two other people. These other participants were following a flexible conversation script and the user's task was to engage in the conversation and, upon receiving a cue, send one of the other participants a private question. During the conversation the user's connection was temporarily interrupted and then resumed by the testers, in an attempt to give observers a chance to see if the user realized what had happened.

The second part of the experience started out with a NetMeeting chat screen not showing the edit window. The other two participants' role was to engage in lively conversation about the user's hometown, purposefully making some untruthful statements about the place. The users' task was to join the conversation and react to these statements, providing correct information about their hometown. Although the task description did not explicitly ask the users to open the edit window, they would obviously have to do so in order to send the messages.

The goal of this case study was to find out if very novice users would be able to use a textual chat, and to identify the types of communication breakdowns (if any) these users would have. It was interesting to notice that none of the users were able to immediately identify the phone ringing sound in the computer as an incoming call. When they eventually did, they had difficulty in finding out how to answer it. They also had a hard time keeping up with the conversation, probably because it took them a very long time to type their messages. As a consequence, they would loose track of the ongoing messages exchanged by the other participants, and their intervention would be typically *late* and off the current topic. The test collaborators commented to the observers that sometimes they would even think the users were having connection problems, since they were silent for so long. Moreover, none of the participants realized they had been disconnected for a while.

Our interpretation of this experience was that although participants were able to exchange messages, they were actually unable to take part in the ongoing conversation. The breakdowns observed in this case study have pointed out to the need to consider problem categories related to network connection and the group's discourse.

Case Study 2 - System Developers

This case study was carried out with two groups, with three participants each. The participants were all system developers who had rarely (if ever) been in chat rooms before, although some of them were frequent ICQ users. All of them were regular (not necessarily advanced) Windows users. Each group's task was to work through NetMeeting and come to an agreement about the spatial configuration of a computer science department's lab. Each user was given a different lab configuration, as well as a set of reasons he or she would use to argue for it. They were told to explain their initial proposal to the others using the chat and whiteboard, but as the discussion with others went on he or she could

be convinced that some other proposal was better.

The goal with this test was to find out if users could carry out a collaborative task using the chat and whiteboard. NetMeeting does not relate the user's cursor in the whiteboard with his/her identification in the chat, nor does its whiteboard discriminate among users' cursors. Consequently users ran into some interactive problems in trying to communicate through both means and coordinate their activities.

Our interpretation of this experience was that although participants were experienced in using technology and communicating through other kinds of software applications such as mail and instant messengers, they had difficulties in interacting through it to solve a problem. Some of the breakdowns we have identified are related to how the group organizes itself and how it makes sense and use of the different communication channels available (in this case, chat and whiteboard). They are also related to how it keeps track of the thread of conversation. As expected breakdowns were identified both at the individual level (user-system communication), and the group level (user-user communication). It is noteworthy that, in our observation, breakdowns at these two levels systematically had consequences for group interaction as a whole.

In the next section we present the levels of breakdowns that we have identified in these case studies, as well as some examples of each of them.

Levels of Breakdown in Groupware Interaction

As in single-user applications, when a participant is interacting with the application user-system communication breakdowns may take place. Although these can be characterized by the original set of utterances, it is interesting to notice that they bring in a new dimension: its consequences for the group. When such breakdowns happen the participant has to interrupt his group communication/activity to take care of the problem. This may cause him to lose track of the communication thread and/or interrupt his actions regarding group activity. Thus, although other participants may not be aware of this participant's difficulties, they may notice the interruption in his activities. This may cause further interaction breakdowns at group level, since the group may have to wait for the participant to get around the problem before they can go on with their common activity. Another possibility is that upon his return the participant could attempt to pick up the conversation from where he had left off and this could be disrupting for the group.

In our case studies we identified all such breakdowns. A good example of individual breakdowns happened in group 1 of case study 2. Participant 1 of this group suggested that they used colors to be able to recognize each other while using the whiteboard. Participant 2 said he would use red and went on to setting his color to red. In the process of doing that he had a number of breakdowns at the individual level (e.g. couldn't find the pen color tool). While he was trying to solve the problem, the other two participants went on drawing their proposals. By the time Participant 2 managed to set his color to red, the others had finished drawing. So, when he started drawing the others had already started discussing their proposals. In other words, he was out of phase with the rest of the group. This had some important consequences later on. First the other two carried on the group activity without Participant 2, and at one point they cued Participant 2 to describe his proposal. However, Participant 2 was focused on his drawing and did not perceive the cue. The others then started guessing what could be the meaning of Participant 2's drawing. When Participant 2 eventually finished his job, he immediately typed a message describing his idea without reading or even looking at the conversation that had been going on. In this case, the breakdowns in the individual level interfered with the group coordination and communication, causing some considerable breakdowns at the group level.

At the group level interactive breakdowns are characterized as those that involve one or more members of the group. We have focused on breakdowns that are related to the use of the application as a means of communication, as opposed to problems that may happen in face-to-face situations and are transported to the technology. For instance, while using the whiteboard in case study 2, participants would ask who was drawing, since there was no identification and they had no other way of finding out.

One of the challenges for groupware evaluation is that some of the problems that may occur in this environment may go unnoticed even for the members of the group. However, for an evaluator who is carefully examining the recorded session, some of these problems stand out as crucial,

because they may, for example, lead to false agreements and faulty decision-making. In order to illustrate this point, let's take group 2 of case study 2. Participants 2 and 3 are discussing Participant 2's proposal and they refer to the proposal Participant 3 had shown earlier. Then Participant 2 starts typing a question regarding Participant 3's proposal. While he is typing, Participant 1 sends in a new proposal. Participant 2's question appears in the chat window right after the new proposal is shown, and looks like it refers to Participant 1's proposal instead of Participant 3's. This causes the group to shift to the new proposal topic without having discussed the previous proposal presented by Participant 3. The members of the group did not seem to notice this accidental shift in the discourse topic (not even Participant 2), which brought about serious implications for the group's negotiation process and decisions.

We have classified the types of problems that happen at the group level into four different categories. In the next section we describe each one of these types and present examples observed regarding each one of them.

Types of group breakdowns

In groupware applications new categories of problems specific to group interaction and technology must be added to the existing communicability evaluation categories in order to characterize the breakdowns that may occur in this particular environment. Based on our case studies we propose four types of interactive breakdowns for groupware application: workspace awareness, discourse awareness, technology awareness and coordination awareness.

When working in groups, in order to communicate, collaborate and coordinate with others members of the group, users need information about the other members of the group, as well as their interaction with the application. This kind of information has been called workspace awareness [Gutwin & Greenberg 1996, Gutwin 1997] and its importance in group interaction has been investigated [Gutwin et al. 1996, Gutwin & Greenberg 1998, Gutwin & Greenberg 1999, Baker et al. 2001].

In our case studies, we came across some examples of interactive breakdowns that illustrate this type of problems. For instance, as we have pointed out, NetMeeting whiteboard does not make a distinction among the various users. Thus, in our case studies, at times participants would not be able to identify who was drawing on the whiteboard and explicitly had to ask in the chat. Another example happened in group 2 of case study 2, when Participant 1 was drawing on the whiteboard, and Participant 3 asked him to textually describe his proposal. He stopped drawing and started typing the description. Before he finished, he was interrupted by some messages from the application, and had to deal with them before he was able to get back to what he had been doing. The other participants were waiting for his description, and from their perspective Participant 1 had become quiet and they didn't know why (they didn't know that Participant 1 had started describing his proposal, nor that he had been interrupted). So, they changed the subject and started talking about something else. By the time Participant 1 was able to get back to typing his proposal, he noticed the others had changed subject and erased what he had started typing, without having had the chance to explain it to the others.

In face-to-face communication people can look at the other participants in the conversation and have cues on each participant's partaking in the whole process. For instance, one can notice that: (a) someone is saying something, and he waits until that person has finished; or (b) someone is trying to say something, and he passes the floor to this person; or (c) someone is saying something and someone else nods, and he infers the person agrees with the speaker. All this kind of knowledge about communicative processes is used by participants of a conversation when deciding what to do next and which topic to address. If the necessary cues to activate this knowledge are not provided by groupware applications, then participants may be unable to act appropriately in this communicative setting (or may be forced to try to find a way around difficulties by negotiating social protocols in real time during interaction). We have classified this type of information as discourse awareness.

NetMeeting chat provides users with the information of how many people are participating in a conversation, who they are, and what each one has said, although not all equally accessible. The messages are displayed in temporal order, according to when they were sent. In our case studies, the lack of discourse awareness caused different communicative breakdowns, such as interruption in the conversation flow, participants talking at the

same time about different things, participants not having the chance to say something or even misinterpretations to what the message referred to.

In group 2 of case study 2, at times we could notice long pauses (0.5 to 1 minute long) in which all of the participants would just wait. One of the reasons for this probably was that everybody was waiting for everybody else to say something. Moreover, not knowing that someone was about to say something (typing in a message) also caused users to "talk at the same time". In group 2, at one point the group is talking about the representation of the lockers drawn by Participant 2. Then Participants 1 and 3 send a message at just about the same time. Whereas Participant 1's message is about the other representations in the drawing (e.g. desks and chairs), Participant 3's message discusses the purpose of the lockers. Participant 2 receives both messages almost at the same time and replies with an ambiguous message that could be directed to either one of the two messages received. Participant 1 and 3 regarded the reply as being directed to their own question and continued talking about different topics. The fact that Participant 1 and 3 did not know that the other was saying something (typing a message) caused them to talk at the same time. Moreover, the fact that they did not know to whom Participant 2's reply was directed caused them to continue talking about different topics and the group conversation was divided into 2 different threads, each person taking part in only one of them. In this case, both participants sent their messages and continued talking. However, it was often the case that when a participant finished typing and noticed that someone else had just sent in a message changing topics, he just erased his message and addressed the new topic in a new message. In an extreme situation this could cause the participant never to be able to talk, or just to be disruptive talking about topics that were not being discussed any longer.

We have called the third category technology awareness, encompassing breakdowns that take place due to how much knowledge the user has of the underlying required technology for groupware applications. Although ideally the user should not need to know the details about how the technology works in order to use it effectively, in groupware applications this knowledge may be useful or necessary to interpret the signs or messages present in the application. For instance, as described above the participants of case study 1 were novices and had no knowledge of what it meant to be connected to the Internet. Thus, when their connection was interrupted they could not understand what had happened, nor could they come up with a reasonable explanation of why the others wouldn't answer any longer. For example, when faced with this situation, Participant 2 kept asking the others where they had gone. On the other hand, knowing the technology can also be the cause of some communicative breakdowns. For instance, the collaborators of this same case study commented that at times they had thought that the connection was down, since the novice took so long to reply. In fact all that happened was that the novice was having difficulties in keeping up with the conversation. These problems show that a good and reliable representation of how the necessary technology is functioning, particularly the connection, has to be available to users, so that they are able to make sense of some of the signs that are or should be perceived as associated to it.

Coordination of the group's activities is crucial to group performance. When the application does not provide tools for the group to coordinate its activities, this may have a negative impact on the group's efficiency. Thus, we have identified the fourth group problem category as coordination awareness. The fact that the whiteboard did not make the distinction among the users, made it hard for the group to organize itself. Group 1 in case study 2 soon realized the problem and defined a way to get around the problem by assigning a color for each participant. Group 2 did not realize this, and members were unable to draw and present their proposals to others. Besides, in group1 each participant was drawing on a different part of the whiteboard, which allowed them to see all proposals simultaneously. Group 2 did not use this spatial dimension to differentiate among the various proposals, and subsequent changes to the same drawing led them to differentiate proposals only temporally (i.e. who drew first, next, and so on). This is a much more difficult strategy than the one followed by group 1, because it imposed much heavier cognitive loads on participants. One of the communicative breakdowns we observed in group 2 was, for instance, that the first participant to draw a proposal was startled when another participant started making changes to what had been drawn. In this participant's perspective, the drawing was hers, and not the group's. Another breakdown occurred because participants could not see all proposals at the

same time, so one participant ended up drawing his proposal twice.

Conclusions

The lack of practical methodologies to evaluate groupware applications [Baker et al. 2001] calls for research and development of such methods. In this paper, as a first step towards extending the communicability evaluation method to groupware applications, we have proposed four categories to characterize communicative breakdowns that may occur in groupware applications: workspace awareness, discourse awareness, technology awareness and coordination awareness.

Workspace awareness has been proposed by Gutwin and Greenberg [Gutwin & Greenberg, 1996, Gutwin 1997] and has been used as basis to research regarding design and evaluation of groupware applications [Baker et al. 2001]. It provides members with information about other members' interaction with the application, and thus supports their decisions regarding their next actions and their collaboration with others. Discourse awareness is needed to yield group members with useful and necessary cues that will allow them to communicate effectively. By communicate we mean not only exchange messages, but also achieve an intent through this message exchange and have a consistent and cohesive discourse. When interacting with a group through a software application, it is often useful for users to have some knowledge regarding the underlying technology, so that they can make sense of signs and messages associated to it. Technology related signs that are available to users explicitly (e.g. indication whether they are connected or not to the others) or implicitly (e.g. time interval between the display of two messages at the chat window, that may be used by the user to infer that a remark is not related to the one presented immediately before it in that window) contribute to their technology awareness and improve their interaction with the software and the group. Finally coordination awareness is associated with providing users with the knowledge of which protocols are available to support their interaction and of how they can be useful.

One could question the need for four different categories, as opposed to just extending the concept of workspace awareness or creating a catch all group awareness category. We would reply that since the knowledge and problems involved in each one of them are of a different nature, making this distinction clear can help designers pinpoint interactive problems, as well as allow them to refine their decision-making regarding what kind of information they should provide users with. So, even though this is just an initial step towards extending the communicability evaluation method, it can already contribute to the design of groupware applications.

One other important result of our investigation in this paper that should be considered by designers is the impact of individual interactive breakdowns at the group level. The breakdowns at the individual level are equivalent to those that may take place in single-user applications. However, in groupware applications they often cause group breakdowns as well. Although we have established this causal relationship between breakdowns at individual and group levels, we would like to explore this relation further and investigate whether there is a hierarchical structure of breakdowns. That is, examine if specific breakdowns at the individual level always yield the same breakdowns at the group level.

Further investigation and research is also needed in order to extend the communicability evaluation method for groupware applications. In this direction, we must work towards defining the set of utterances that describe group breakdowns. Once this set of utterances has been defined we must investigate how the tags can be associated to the problem categories we have proposed in this paper. We would then be in a position to perform tests that would allow us to assess the categories proposed in this paper and have stronger evidence of their adequacy. The following step would be to integrate this extension proposal to the original method, identifying how the new set of utterances and problems relate to the original ones. Finally we would have to perform extensive testing to evaluate the extended method being proposed and compare it to existing methods.

References

[Baker et al. 2001] Baker, K., Greenberg, S. and Gutwin, C. (2001) Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration. *Proceedings of the 8th IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI'01).* (May 11-13, Toronto, Canada).

[Blythin et al 1997]. Blythin, S., Hughes, J. A., Kristoffersen, S., Rodden, T. and Rouncefield, M. (1997) Recognising 'success' and 'failure':evaluating groupware in a commercial contex. *In the Proceedings of GROUP 97*, Phoenix, Arizona, 39-46.

[Cugini et al 1997] Cugini, J., Damianos, L., Hirschman, L., Kozierok, R., Kurtz, J., Laskowski, S. and Scholtz, J. (1997). Methodology for Evaluation of Collaboration Systems. *The evaluation working group of the DARPA intelligent collaboration and visualization program, Rev. 3.0.* http://zing.ncsl.nist.gov/nist-icv/documents/method.html

[de Souza, 1993] de Souza, C.S. (1993) The semiotic engineering of user interface languages. *International Journal of Man-Machine Studies* 39 (1993), pp. 753–773.

[de Souza et al. 1999] de Souza, C.S., Prates, R.O., and Barbosa, S.D.J.(1999) A Method for Evaluating Software Communicability. *In the Proceedings of the Second Brazilian Workshop in Human–Computer Interaction* (IHC '99).

[de Souza et al. 2000] de Souza, C.S., Prates, R.O., Barbosa, S.D.J..(2000) A Semiotic Engineering Approach to User Interface Design. Available via http://peirce.inf.puc-rio.br/ . Revised version to appear in Knowledge-Based System, Elsevier, 2001.

[Eco, 1976] Eco, U. (1976) A Theory of Semiotics. Indiana Univ. Press, 1976.

[Gutwin 1997] Gutwin, C. (1997). Workspace awareness in real-time distributed groupware. Ph.D. Dissertation. University of Calgary, Calgary, Canada. Available via http://www.cs.us-ask.ca/faculty/gutwin/publications.

[Gutwin & Greenberg, 1996] Gutwin, C. and Greenberg, S. (1996). Workspace awareness for groupware. In *Proceedings of the CHI '96 Conference Companion on Human Factors in Computing Systems: Common Ground* (CHI '96, Vancouver, British Columbia, Canada, Apr. 13–18), M. J. Tauber, Ed. ACM Press, New York, NY, 208–209.

[Gutwin & Greenberg 1998] Gutwin, C. and Greenberg, S. (1998). Effects of awareness support on groupware usability. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI '98, Los Angeles, CA, Apr. 18–23), M. E. Atwood, C.-M. Karat, A. Lund, J. Coutaz, and J. Karat, Eds. ACM Press/Addison-Wesley Publ. Co., New York, NY, 511–518.

[Gutwin & Greenberg 2000] Gutwin, C. and Greenberg, S. (2000). The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. *IEEE 9th Int'l Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE'00).*

[Gutwin et al. 1996] GUTWIN, C., ROSEMAN, M., AND GREENBERG, S. (1996). A usability study of awareness widgets ina shared workspace groupware system. In *Proceedings of the 1996 ACM Conference on Computer-Spported Cooperative Work* (CSCW '96, Boston, MA, Nov. 16–20), G. Olson, J. Olson, and M. S. Ackerman, Eds. ACM Press, New York, NY, 258–267.

[Hartson 1998] Hartson, H. R. (1998) Human-computer interaction: Interdisciplinary roots and trends.*The Journal of System and Software,* **43**, 103-118.

[Norman 1986] Norman, D. Cognitive Engineering. In Norman, D. A. and Draper, S. W. (eds), *User-Centered System Design.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1986, pp. 411–432.

[Norman 1988] Norman, D. (1988) *The Design of Everyday Things.* Doubleday/Currency Edition. New York.

[Prates et al 2000a] Prates, R., de Souza, C. and Barbosa, S. (2000) A method for communicability evaluation of user interfaces. *Interactions*, ACM, **7**(1), 31-38.

[Prates et al. 2000b] Prates, R.O.; Barbosa, S.D.J.; de Souza, C.S. (2000). A Case Study for Evaluating Interface Design through Communicability. *In Proceedings of the ACM Designing Interactive Systems, DIS'2000* Brooklyn, NY. August, 2000, pp 308316.