Penz, F., P. Antunes and M. Fonseca (1996) Feedback in Computer Supported Cooperation Systems: Example of the User Interface Design for a Talk-Like Tool. The Design of Computer Supported Cooperative Work and Groupware Systems. D. Shapiro, M. Tauber and R. Traunmueller, Human Factors in Information Technology, North Holland. Selected papers from the 12th Schaerding International Workshop, The Design of Computer Supported Cooperative Work and Groupware Systems held in Schaerding, Austria, June, 1993.

Feedback in Computer Supported Cooperation Systems: Example of the User Interface Design for a Talk-Like Tool

Franz Penz [*]
Pedro Antunes [†]
Manuel Fonseca [‡]

Abstract

CSCW systems need to support very expressive kinds of feedback since cooperation requires strong interaction between users. Computer supported meeting rooms solve that problem by mediating video and audio channels and therefore simulating the face-to-face environment. Groupware on a wide area network, however, should rely on different strategies to provide the same degree of interaction mostly because of bandwidth and time delays.

This paper discusses one tool, FaceTalk, that, along with text, communicates graphical faces. These faces can be set up to express feelings like attentiveness, eagerness, friendliness, agreement, disagreement, sadness, irritation, confusion, adding to plain textual messages a strong context of interpretation.

keywords: CSCW Design, Feedback, Wide Area Communication, Talk-Like Communication

1 Introduction

Though communication, whether it is computer supported or not, is only a means to achieve the actual aims of a business it is at the same time an important end in social relations. Communication during the working life, whether formal or informal, is of equal importance for the work itself as well as for work satisfaction of the coworkers. [Ellis 91] even reports that users after having experienced a group editing tool found it frustrating to work with single user tools afterwards.

Now, as we are no longer communicating only with the computer but instead using the computer as a tool to communicate with other human users, we do not only need feedback whether the computer has understood us but also from our human communication partners.

Because of the small bandwidth of usual text or even graphics based computer communication, additional information, especially in same time/same place environments, is mediated "out of band" by extra voice and video lines or simply face to face (e.g. in special meeting room installations like [EK90]).

The application of these extra channels may not be easily extended to wide area cooperations because of two main reasons: it is too expensive for wide area cooperations today and the delays in communication are high enough to impose different interaction styles between users. New kinds of feedback should be explored, expressive enough to improve cooperations and suited for living with wide area network limitations.

In this paper we discuss some enhancements of the feedback provided to users using talk-like communication tools and present a tool, FaceTalk, which adds drawn face expressions to a talk session. Because we know the problems of assigning CSCW related work in this still very young field we start in section 2 with a classification of FaceTalk along several technological CSCW dimensions. Section 3 deals with the current problems of CSCW applications and also gives a motivation for designing one. In section 4 we discuss feedback in the context of CSCW. In sections 5 and 6 we describe our requirements for a talk-like communication tool and our actual design of FaceTalk. We close with our experience and expectations on the use of the first FaceTalk prototype.

2 Design of FaceTalk in different CSCW dimensions

The most used dimensions to classify groupware are space and time (e.g. [Rodden 91,Ellis 91, Rodden 92,Hiltz 91]). Regarding the form of the cooperation, applications may support cooperative work of users who simultaneously work with the system (synchronous systems or real-time groupware) or without the simultaneous presence of all

group members (asynchronous systems or non-real-time groupware). Of course an application may have a mixed form supporting both. Regarding the geographical nature of systems the two extreme positions are remote and co-located.

As FaceTalk shall support interactive communication as well as features known from phone answering/mailing systems, FaceTalk supports a mixed synchronous/asynchronous cooperation form. FaceTalk is intended to be used remotely over a wide area network. Transmission delay times are a problem for interactive communication. For instance, we could measure mean delay times up to three seconds between Portugal and United Kingdom when the net was working well. It is therefore important to incorporate into an application adequate feedback on what has really been received on the other side.

A further common classification is based on the technical implementation of the cooperation. Collaboration-aware applications are designed to support cooperation while collaboration-transparent applications are normal single user applications used in a collaboration task. By means of support from a distributed system or a shared windowing system every existing single user application can be used in a cooperative way [Cockburn 91,Lauwers 90].

The FaceTalk design is a collaboration-aware application by its nature. Many of its features emerge from collaboration considerations in course of the user interface design.

Some authors introduce further dimensions enriching this simple classification scheme:

Malone et al. [Malone 87] classify according to the formal knowledge the systems contain on the application domain supported. Forms processing and calendar management are examples of applications containing a great deal of formalized knowledge, electronic mail, computer conferencing or hypertext systems are examples with very little domain knowledge.

Some semantic knowledge is a requirement for providing additional feedback. FaceTalk "knows" different communication elements (faces, talk text, out-messages, ... - see below). Still, as it is a talk tool, it rates low in the domain knowledge dimension.

Rodden and Blair [Rodden 91] define five classes of CSCW systems based on the representation of control they embody, ranging from explicit to implicit control: speech act or conversation based systems, office procedure systems, semi-formal active message systems, conferencing systems, and

peer-group meeting or control free systems. Explicit control systems provide means to tailor group interaction and cooperation whereas implicit control systems do not.

Actually, the existence of formal knowledge seems to be highly correlated with the possibilities of controlling the group interaction and cooperation.

FaceTalk provides only few mechanisms to control users' interactions and therefore is an implicit or control free system. As is with most of such systems, FaceTalk relies on social protocols to establish appropriate forms of coordination.

Ellis, Gibbs and Rein [Ellis 91] give dimensions for a spectrum of systems being groupware more or less. "Common task" regards whether the users perform their tasks separately and independently or whether they are closely interacting on the same task. A "shared environment" dimension describes the extent to which the system provides information about the participating people, the ongoing project, and so on.

Their taxonomy of CSCW systems uses the application-level functionality in addition to the time/space matrix. As a non-comprehensive set of classes they suggest message systems, multi-user editors, group decision support systems and electronic meeting rooms, computer conferencing (which is further divided into real-time computer conferencing, computer teleconferencing, and desktop conferencing), intelligent agents, and coordination systems.

FaceTalk users are closely interacting on a common (communication) task but rate low in the shared environment dimension because they have only a little more information than with a standard mailing system. At the application level functionality FaceTalk falls into multiple groups: it can be used as a messaging system and as a simple real-time computer conferencing tool.

Nunamaker et al. [Nunamaker 91] use the group size in addition to the time and space dimensions to classify electronic meeting systems.

We want to use the group size to divide between (non groupware) single user applications, two user applications (like U*IX talk), applications for small groups (every one knows every one) and applications for large groups. FaceTalk is a two user- to large group application but only small groups may communicate effectively.

Applegate [Applegate 91] presents a very comprehensive alignment model for CSCW covering most of the dimensions we have already identified. Group, task and technology form the basic dimen-

sions, each of them providing several further perspectives. We only want to stress one more element from this work: the level of support perspective from the technology dimension.

Computer based tools for human cooperation support both communication and collaboration tasks. The level of support perspective allows us to distinguish between these tasks. Actually, cooperation usually requires a lot of both. Because of the nature of computer work – it all deals with information – we cannot draw a hard border line between communication and collaboration. As no strict technical criterion can be applied, the difference between communication and collaboration has to be found in the intentions of the users or the subjective opinion of the observer. In fact, the actual aspect of a cooperation task may switch during execution. Elwart-Keys et al. [EK90] even report on the tendency of computer supported meetings to switch from a "talking about work" style to "doing work."

From its basic structure FaceTalk primarily supports communication tasks.

Table 1 sums up the CSCW dimensions and our classification of FaceTalk.

Dimensions	Classification
cooperation form	mixed (synch. + asynch.)
geographical nature	remote
design	collaboration-aware
domain knowledge	low
control representation	implicit (control free)
common task	high
shared environment	low
app. level function.	message system, real-time computer conf.
group size	small
support level	communication

Table 1: Classification of FaceTalk

3 Design of FaceTalk for CSCW success

As happens in every new field that cannot show early success in satisfying its end users, articles start to appear dealing with "why CSCW fails." Erickson in [Erickson 89] reporting on Grudins paper [Grudin 88] gives invaluable hints on how to design new systems.

Grudin identifies three factors responsible for the failure of CSCW applications: the cost vs. benefits distribution on different group members, the difficulties in the designing and, particularly, the evaluation of CSCW systems.

The design of groupware is not only difficult by itself, the introduction of new tools and features to be used in the group communication process can have unexpected and drastic consequences for the communication process, the information communicated, or the group itself [Erickson 89]. Though this is true for every new bit of software in any environment, the magnitude of the consequences seems to be amplified for groupware because here we have to deal with the behaviour of multiple users instead of a single one and here ...*the whole is more than the sum of the parts...* [Erickson 89].

In our opinion we find the real limitations of groupware design today, especially same time/different place groupware, on the pure technical level of the supporting hardware. The speed of usual non-local networks are far from covering the needs of synchronous group interfaces.

Regarding feedback, manipulations accomplished by users or systems should be shown (fed back) immediately. I.e. the delay between manipulation and output must not be noticeable by humans. As long as this cannot be guaranteed with the available network technology, we have at least to support users with an adequate conceptual model of the system and with feedback telling them what is really going on. FaceTalk gives an example how to tackle this problem by providing a feedback in the talked text showing what is for sure already on the other side (see below).

According to Rodden and Blair [Rodden 91] we may be more or less in a deadlock situation: ...*a major problem in distributed systems is a lack of existing applications of the technology leading to technological solutions to technological problems.*

4 Feedback in CSCW systems

The feedback provided by CSCW systems can be classified into 3 kinds:

- feedback on the interaction with the machine – **feedback from the system**

- feedback from human cooperation partners mediated by the system (in-band) or mediated via a different communication channel outside the system (out-of-band) – **feedback from the users**

- feedback supplied by the system on the actual status of cooperation (e.g. for partners entering a group) – **feedback about others**

Every communication whether human or technical becomes much easier if adequate feedback is

provided. In human computer interaction the benefits of providing any possible feedback has led to the realization of the direct manipulation interaction style in modern user interfaces.

There is no doubt on the relevance of feedback in human-human communication too. Consequently, it is obvious why in group cooperation and therefore in groupware ...*it is very important that each user can see what other users are doing* [Lee 90].

In the course of everyday human interaction, whether face to face or by phone, we use an extremely rich spectrum provided by intonation and/or body language and facial expressions to mediate our feelings [Viller 91] and to synchronize the communication. Especially, gestures are used to focus attention and control the flow of a conversation (turn taking) [Greenberg 89]. Thus, in the long run, we cannot do without voice and video channels supporting our cooperations (as for instance in the TeamWorkStation [Ishii 91] or ARKola [Gaver 91]).

Unfortunately today, a sufficiently powerful network connection is at most available in local sites. The usual wide area connections are too slow to allow the interactive sending of large amounts of additional unstructured data. This condition results in current systems that only provide textual communication. Frequently, however, textual feedback is not obvious to users [Jirotka 91].

Since group collaboration only extends these problems, the feedback about others, i.e. a feedback produced by the machine about people, represents a realistic compromise between expression and bandwidth. Telepointers are for example used to identify activities and sometimes thoughts of cooperating people, menu bars with pictures show group participants, and painted locked areas are used to avoid conflicts.

One concept which has much to do with "feedback from the users" and "feedback about others" is WYSIWIS [Stefik 87] (What You See Is What I See). In a WYSIWIS system users can watch what other users are doing. The system provides feedback about others. Users, being aware that these interactions with the system are observed by others, can use WYSIWIS for communication. This is clearly feedback from the users supported by feedback about others.

Whether to support strict WYSIWIS, forcing the screen image for all users to be identical, allowing them to talk in terms of spatial references [EK90], or to relax the concept, allowing every user to configure the perceived information according to individual needs [Greenberg 90], is actually a question of the chosen metaphors. Providing a conference room metaphor [Lauwers 90] there is clearly no way around strict WYSIWIS.

From the point of view of user interface design we can give a more general solution. As with any software, the relevant aim of a user interface design is to build up the correct conceptual model of the actual application functionality in the user's mind by means of the user interface [Penz 91]. So, what we really want to provide is rather a WYGIWIG (What You Get Is What I Get) with respect to the conceptual models. It is not important that user interfaces of a cooperative application are identical for all cooperating users. What is essential is to suggest identical conceptual models of what is going on to all of them. According to the task at hand WYGIWIG will then force strict WYSIWIS or not.

5 Requirements for interactive talk-like communication

Basing on our own experience using the standard U*IX talk and network news service we identify some features we are missing in the communication process:

First of all some information reporting whether the intended communication partner is at the system or not could be a very useful feature. As many people tend not to logout during the lunch break or if they leave their computer for a longer period, the existing talk cannot provide this information (obviously a feedback about others). In addition, we still have to fight against the sometimes bad connection lines and so we actually cannot know if, perhaps, only the line is not available at some time.

A more sophisticated "The user is out" – or perhaps "I am out" – feedback should not only allow us to know whether users are in or out but also whether users want to be contacted at the moment and when they will be back. Further, the facilities of a common phone answering machine should be provided, allowing also to leave messages to be read when the user returns.

A further requirement rises also because of the sometimes slow connections. In a talk session we really would like to know what is already on the screen on the other side to be able to connect the given answers to our questions and vice versa. To know this exactly is technically difficult, but we can at least provide feedback information about message delays by showing what was for sure received and shown on the screen of the other side.

During talking we want to transfer information on how our messages are to be understood and what our feelings are in response of what we read. These feelings are sometimes hard to put into text quickly in an expressive and unambiguous way. This situation is also very common in newsgroup messages where "smilies" are used to express laugh, irony or sadness.

In addition we frequently would like to convey quickly and simply that we are now speaking or phoning with someone in parallel or simply give standard ok-s, no-s and hello-s. If it is because of a phone call you simply cannot type that much to say "a moment please, I have a phone call..."

The standard talk is very limited in providing history of the exchanged text. Not only would it sometimes be useful to have a protocol of a talk, but sometimes because of a slow connection it may happen that too many lines arrive at once. Then, parts of the text are lost because the window is filled in a cyclic way, overwriting the oldest part. Therefore it is also not possible to just cut and paste more than very few lines into the talk window.

Many times it would be interesting to do a group discussion as is possible with xparty [Peebles 92]. It should be possible to join new communication partners to an ongoing talk session.

6   FaceTalk - an integrated communication tool providing face expressions

This section presents the important parts of the user interface of our FaceTalk design.

There is one main window which is on the screen all the time (see figure 1). In this window you can edit messages and set your faces. There is also a button which opens a phonebook window. The phonebook lists previous connections, allows to enter new addresses and has a button to establish
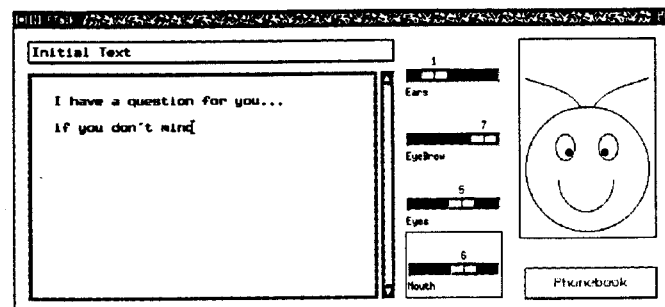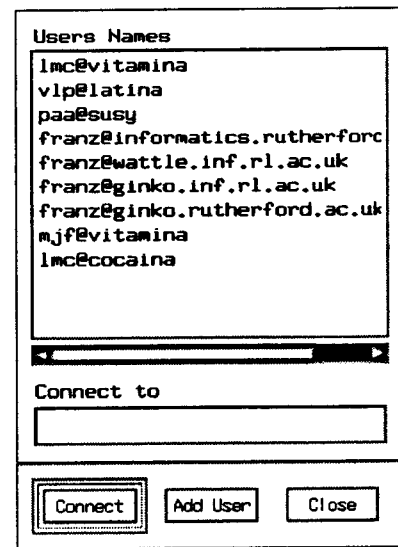
connections (see figure 2).

Figure 2: FaceTalk phonebook window

When establishing a connection a request window pops up on the receiver's side. That window shows your introductorial message and face and allows to accept the connection, refuse it or not to be there (selected automatically after a short delay). See figure 3.
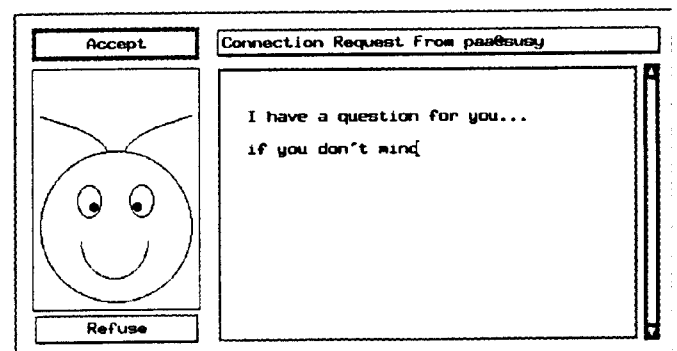

Figure 3: Sample of a FaceTalk request window

If the connection is established each partner will have one FaceTalk main window allowing to enter text and to set faces, and one similar receiving window for every other partner, see figures 1 and 4.

If the talk connection is not established your introductorial message is left at the receiver's side.

At the moment the faces are specified by manipulating four different characteristics: eyes, ears, mouth and brows. For each one of these elements a scrollbar allows you to choose from several options, e.g. eyes more or less open. You can see some possible faces in figure 5.

Our actual intention is to supply scrollbars parameters of the mood instead of face elements. The user could then choose between different degrees of
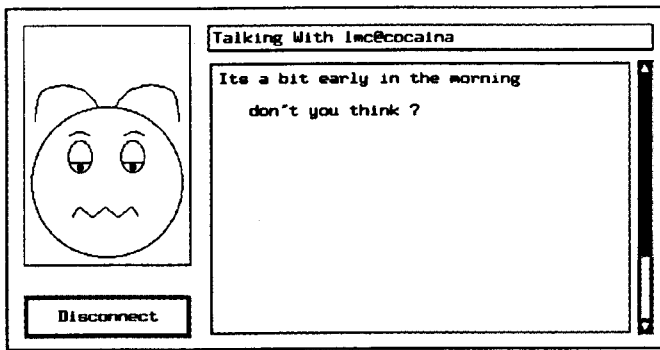

Figure 1: FaceTalk main window

Figure 4: Sample of a FaceTalk receiving window

being happy or sad, active or tired and so on. But, as we do not know what feelings will be relevant and how to map them to the face elements we decided to defer the feeling-to-face translation to a later prototyping step. We first want to collect some experience in possible face modification and its application, i.e. what kind of face expressions are actually used and in what form and context.
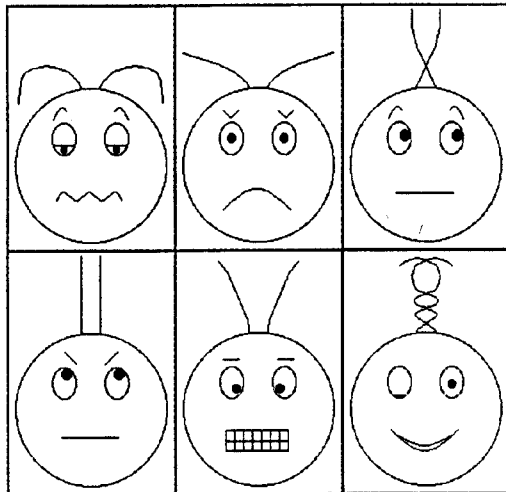

Figure 5: Some possible faces

There are two more windows in FaceTalk that are not shown here: one out-message window, a simple edit field to write the text that appears automatically on the other side when you receive a talk request and press the Refuse button, and one history window containing text of old talk sessions and messages which arrived while you were absent.

7 Experience with FaceTalk

After implementing the first FaceTalk prototype we are especially interested in getting experience in the actual use of the faces (feedback). We are testing how much it is used and if our user interface to adjust the face parameters is simple enough. We are evaluating which feedback elements are of use

and which are missing. In an iterative process we are adapting the prototype accordingly.

Some of our early results are the following:

- Time delays in wide area communications are the biggest constraint while using FaceTalk.

We noticed that users quickly loose links between sent and received text and faces, the discourse becomes incoherent and they have to stop and make checkpoints.

Therefore it will be very helpfull to know what has been received by the other participants of a FaceTalk session (the current prototype does not support this feature of our design).

As a consequence of time delays users feel to have plenty of time to do some local work. This time could not only be used to look at previous text (as it is allready possible with the current FaceTalk) but also to annotate it or even to structure received messages into classes. An advanced talk application could provide means to support this.

- Usage of faces suffers right now of a "new toy" problem, i.e. users get fun just for using different faces which results in over-usage.

- Users tend to choose the most expressive faces.

While usage of faces will change in time, when people learn some of their subtleties, we might need to provide even more expressive faces.

Further expectations

In spite of being aware of the problems in foreseeing the consequences of using FaceTalk, we try to consider what we may observe during further evaluation:

- The users may not be able to associate a known expression with our primitive face drawings.

- The users may not identify themselves with their own face drawing and therefore refuse to use the additional communication channel provided.

These two problems may possibly be reduced by allowing users to set up automatically an individual basic face and allow further minor configuration features.

On the other hand our faces preserve the anonymity of simple computer-mediated communication systems. Because of this anonymity individuals from traditionally discriminated groups in our

society ...*stand more chance of being treated according to what they say rather than what they look like* [Viller 91].

- The face parameterisation allows more intensive expressions than the ones most people are able to do with their real face.

This possibly might have implications on the communication behaviour which we do not yet know. We may however expect more crude reactions from FaceTalk users than usual in real world face-to-face communication.

- A further problem is the user interface for setting the face expressions. Though we have tried to make it as easy as we could imagine to set up the expression values by providing an as-we-think quick keyboard control, that might not be sufficient. First, it may be even too trying for the user to carry out the additional input compared to our natural face which works more or less automatically. Second, feelings are often reactive "events" occurring during communication.

In the next prototype we will divide the face setup interface into a more static part defining our basic state of today, that can be controlled with scrollbars, and a spontaneous part allowing users to signal spontaneous feelings. If the user presses one button to signal a feeling, the face expression will change from its current state to a state reflecting the user's feeling and then slowly return to its setup of the day.

8 Conclusion

We are starting to explore the usage of graphical faces in wide area computer communication. We think that this kind of feedback improves communication bandwidth since it allows users to quickly express feelings that are sometimes difficult to express by words. We also intend to study new kinds of "feedback about people" in cooperation systems.

References

[Applegate 91] L. Applegate. Technology support for cooperative work: a framework for studying introduction and assimilation in organizations. *Journal of Organizational Computing*, 1(1), 1991.

[Cockburn 91] A. Cockburn and H. Thimbleby. A reflexive perspective of CSCW. *SIGCHI Bulletin*, 23(3), July 1991.

[EK90] M. Elwart-Keys, D. Halonen, M. horton, R. Kass, and P. Scott. User interface requirements for face to face groupware. In *CHI '90: Conference on Human Factors in Computing Systems*, April 1990.

[Ellis 91] C.A. Ellis, S.J. Gibbs, and G.L. Rein. Groupware: Some issues and experiences. *Communications of the ACM*, 34(1):38–58, 1991.

[Erickson 89] T. Erickson. An eclectic look at CSCW '88. *SIGCHI Bulletin*, 20(5):56–64, 1989.

[Gaver 91] W. Gaver. Sound support for collaboration. In *Proceedings of the Second European Conference on Computer Supported Cooperative Work - ECSCW '91*, pages 293–308, 1991.

[Greenberg 89] S. Greenberg. The 1988 conference on computer-supported cooperative work: Trip report. *SIGCHI Bulletin*, 20(5):49–55, 1989.

[Greenberg 90] S. Greenberg. Sharing views and interactions with single-user applications. In *Proceedings of the ACM/IEEE Conference on Office Information Systems*, pages 227–237, Cambridge, Massachusets, 1990.

[Grudin 88] J. Grudin. Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '88)*, pages 85–93, Portland, Oregon, 1988. ACM Press.

[Hiltz 91] S. R. Hiltz, D. Durfner, M. Holmes, and S. Poole. Distributed group support systems: social dynamics and design dilemmas. *Journal of Organizational Computing*, 2(1), 1991.

[Ishii 91] H. Ishii and K. Arita. Clearface: Translucent multiuser interface for teamworkstation. In *Proceedings of the Second European Conference on Computer Supported Cooperative Work - ECSCW '91*, pages 163–174, 1991.

[Jirotka 91] M. Jirotka, P. Luff, and N. Gilbert. Participation frameworks for computer mediated communication. In *Proceedings of the Second European Conference on Computer Supported Cooperative Work - ECSCW '91*, pages 279–291, 1991.

[Lauwers 90] J.C. Lauwers and K.A. Lantz. Collaboration awareness in support of collaboration transparency: Requirements

for the next generation of shared window systems. In *Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing*, Seattle Washington, 1990. ACM Press.

[Lee 90]   J.J. Lee. Xsketch: a multi-user sketching tool for X11. In *Proceedings of the Conference on Office Information Systems*, pages 169–173, Boston, 1990.

[Malone 87] T. W. Malone, K. R. Grant, K-Y. Lai, R. Rao, and D. Rosenblitt. Semi-structured messages are surprisingly useful for computer-supported coordination. *Acm Transactions on Office Information Systems*, 5(2):115–131, 1987.

[Nunamaker 91] J. Nunamaker, A. Dennis, J. Valacich, D. Vogel, and J. George. Electronic meeting systems to support group work. *Communications of the ACM*, 34(7), July 1991.

[Peebles 92] A. Peebles. xparty. Appeared on comp.sources.x newsgroup, 1992. peebles@mips.com.

[Penz 91]   F. Penz, M. Manhartsberger, and M. Tscheligi. The world of objects - a visual object based interaction language. In *Proceedings of the 10th Interdisciplinary Workshop on "Informatics and Psychology"*, Schaerding, Austria, May 1991. To be published.

[Rodden 91] T. Rodden and G. Blair. CSCW and distributed systems: the problem of control. In *Proceedings of the Second European Conference on Computer Supported Cooperative Work - ECSCW '91*, 1991.

[Rodden 92] T. Rodden. ECSCW'91 developers workshop. *ACM SIGOIS Bulletin*, 13(1), April 1992.

[Stefik 87] M. Stefik et al. Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. *Communications of the ACM*, 30(1):32–47, 1987.

[Viller 91] S. Viller. The group facilitator: a CSCW perspective. In *Proceedings of the Second European Conference on Computer Supported Cooperative Work - ECSCW '91*, pages 81–95, 1991.