INTERFACES FOR COOPERATIVE WORK: AN ECLECTIC LOOK AT CSCW '88

THOMAS D. ERICKSON

Computer Supported Cooperative Work (CSCW) is one of the newer hot topics in the computer world. CSCW is also known as Collaborative Computing, and CSCW applications are known as 'Groupware' or 'Group Decision Support Systems' (GDSS). The marketplace is just beginning to be subjected to a flood of CSCW applications. It requires no prescience to predict that most of this groupware will fail; as the papers covered in this article demonstrate, there are a lot of uniquely hard problems associated with CSCW that are only just beginning to be understood, let alone solved. Just as one industry wag noted that 'the year of the LAN lasted from 1983 through 1988,' it seems likely that any 'year of groupware' will be good for another five or ten years.

CSCW is of interest to those involved in human interface work for two reasons. First, with the increasing connectivity of the microcomputer environment, and the basic cooperative nature of most tasks, groupware is in everyone's future. Second, while there are certainly technological limitations that need to be overcome, the primary barrier to successful CSCW is a plethora of unsolved human interface problems.

Typical CSCW applications consist of various combinations of the functionality of electronic mail, video conferencing, project management, multi-user databases, and automatic scheduling. (Of course, now that CSCW is a hot topic, just about any piece of software whose design shows the slightest awareness of groups is being touted as groupware–however, this article will focus on full-featured groupware.) The ultimate goal of CSCW is to produce a software environment which will allow people to effectively work together without being physically together. A shorter term goal of some CSCW systems is to provide effective support for groups which are meeting together in person.

This article describes the current state of CSCW–from the perspective of an interface designer–by way of an eclectic summary and discussion of the CSCW '88 conference, held in Portland in late September. Although it doesn't reflect the structure of the conference, in the interest of coherence I will divide the discussion into three sections: problems with existing systems; real world case studies; laboratory prototypes.

PROBLEMS WITH EXISTING SYSTEMS

The consensus of conference participants was that most attempts at CSCW outside the lab have failed. A number of papers were devoted to reviewing various reasons for this failure. It was also pointed out that, of the basic technologies on which CSCW is based, only electronic mail has yet had any measure of success in the marketplace. This section begins with a discussion of Grudin's paper outlining general reasons why CSCW fails. We then turn to a panel which discussed and analyzed "The Coordinator," possibly the most commercially successful of current CSCW applications.

Why CSCW fails. One of the best papers of the conference was Jonathan Grudin's analysis of problems in the design and evaluation of CSCW applications.[1] Grudin identifies three factors responsible for the failure of CSCW. First, there is often an uneven distribution of the costs and benefits of an application among the members of the group it serves. A good example of this is the automatic scheduling application. The idea is simple: users keep their schedules in on-line electronic calendars, and meetings are scheduled simply by selecting a set of participants and allowing the application to find a time and meeting place convenient for all. But, in spite of the attractiveness of this idea, and in spite of the fact that the technology to support such an application has existed for some time, most automatic scheduling applications have failed. A problem is that for such a system to succeed, everyone must use it. This problem is compounded by the fact that those who benefit–primarily managers–are not those who do most of the work. Managers benefit because it's easy to quickly schedule the meetings which constitute such a large proportion of their jobs; but there is much less benefit to the non-management employees who spend less time in meetings. Furthermore, while managers generally have administrative assistants to maintain their calendars, non-managers generally have to maintain their own. Grudin claims that this sort of cost/benefit disparity is a key reason for the failure of many CSCW applications.

Second, CSCW applications fail because intuitions regarding them seem to be uniquely bad. It's hard enough to design a regular, single-user application. Though experienced human interface designers know that their intuitions can't be relied on to assess the difficulties of novices, they have at least some small amount of success at putting themselves into the novice's shoes. Furthermore, they know that their intuitions have some validity for some subset of expert, computer-oriented users. But with CSCW applications, groups are the users. The hackneyed phrase that the whole is more than the sum of the parts surely applies here. To properly anticipate the usability of a CSCW application, the designer must not only understand the individual factors, but must also properly assess factors such as the social structure, politics, and dynamics of the group. Intuition doesn't seem to cover this domain, as evident from the repeated instances of poor CSCW design decisions documented throughout the conference. Some of these will be summarized in the next section.

1. Jonathan Grudin, Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. *Proceedings of the Conference on Computer-Supported Cooperative Work*, pages 85-93, September 26-28, 1988, Portland, Oregon. Also see Jonathan Grudin, Perils & Pitfalls, *Byte*, December, 1988, pages 261-266.

If it were only that design of CSCW is difficult because intuition fails, extensive evaluation of the developing application could compensate for the design difficulty. However, the difficulty of evaluating CSCW applications is the third reason for CSCW failures. How can a groupware application be adequately tested before release? Handing off a CSCW application to a software testing group just won't do the job. The application must be tested by real workgroups, with their own preexisting tasks, social structure, politics, and methods of interaction. Unfortunately, even such testing may not guarantee an adequate product, since such group variables are likely to vary widely between different groups.

The Coordinator. We now turn to a specific example of a CSCW system–The Coordinator. The Coordinator is an interesting and unique application. First, The Coordinator is a commercially available system, now in its second generation. Second, it is based on a theoretic perspective–language as action.The designers had a very sophisticated understanding of language, and how people work together. Third, the program is cleverly designed, and offers a set of seemingly valuable functionality that is not available through any other product. Finally, while it is not yet clear if The Coordinator is a commercial success, the company that produces it (and only it, as far as I know) is still in business.

An exercise for the reader: As you read the description of The Coordinator, try to apply Grudin's analysis of why CSCW applications fail to it. Can you anticipate any of the problems which, with 20-20 hindsight, may appear obvious?

The Coordinator is based on the language-as-action perspective developed by Terry Winograd and Fernando Flores.[2] In this view, cooperative work is a result of a network of commitments which are negotiated and expressed through a few basic units of conversation. Thus, the production of a status report might be a result of the following 'conversation:' A manager *requests* a status report by a particular date; the employee *counters* that the date is too early and suggests a later date; the manager *accepts* the proposed date; the employee *commits* to producing the report; the employee *reports completion* of the status report is completed; the manager agrees and *declares completion* of the report.

This is an instance of what Winograd and Flores call a conversation for action. (Note that these need not be literal conversations; a conversation for action may be spread out

2. See T. Winograd and F. Flores, *Understanding Computers and Cognition: A New Foundation for Design*, 1986. Also see Terry Winograd, Where the Action Is, *Byte*, December 1988, pages 256A-258.

over several days and conducted via email or some other medium.) In short, cooperative work gets done as a result of people making (and negotiating, and renegotiating) commitments to one another. The Coordinator facilitates cooperative work by making these commitments explicit, and by keeping track of what has been committed to and who it affects.

The Coordinator is a combination of an electronic mail system and a project management system. The Coordinator also 'understands' the basic units of conversation–eg. requests, counteroffers, commitments–and what they imply, and it also 'understands' time and its role in cooperative work (eg. deadlines). When a message is sent via The Coordinator, the user specifies the type of message it is (eg. request, commitment, etcetera), in addition to the usual attributes of an email message. Because The Coordinator is also used to keep track of the schedules of projects going on within a company, when someone sends a message which affects the schedule in some way, The Coordinator automatically adjusts the schedule and notifies the people who will be affected. Thus, if a programmer notifies a manager that a completion date for a program module has slipped two weeks, The Coordinator will recalculate the schedule of the entire project, and notify affected team leaders of the change.

On paper, the Coordinator sounds like a manager's dream. It does the drudgery of updating schedules and notifying everyone of changes. There are no conceptually new software applications to learn. The regular employees have only to use the Coordinator as an email system, something with which many are already familiar. Project managers (or their assistants) also have to use The Coordinator as a project management system–but again, this isn't a novel type of application either.

The Coordinator was described and evaluated in a panel presentation at CSCW '88. The panel included Terry Winograd, one of the proponents of the language-as-action perspective, and three evaluators (researchers or managers) who had used and/or studied the use of The Coordinator in commercial settings. Between them, the evaluators had looked at roughly fifteen groups that had used The Coordinator.

Two of the three evaluators had found that there were generally negative opinions. One found that his evaluation group refused to use The Coordinator for the full evaluation period (6 months), the refusal taking on sufficient fervor that the software was, in some instances, literally thrown out of the employees work cubes. The second evaluator, who had looked at the use of The Coordinator in about a dozen work groups, had found that the majority of the groups were unhappy with the product. Generally the unhappiness was directed at the degree of explicitness of The Coordinator: messages had to be labeled according to their 'type'; deadlines had to be set; if someone missed a deadline, even by a little, everyone affected was automatically notified. The word "fascist" was sometimes used to describe The Coordinator. The third evaluator–a manager, user, and strong proponent of The Coordinator–denied these charges vigorously, though he diluted the impact of his denials by remarking that The Coordinator was extremely valuable for "forcing compliance". In general, The Coordinator seemed to work out best in cases where the corporation in which it was used was a very traditional one with a clearly defined hierarchy.

In the cases where it is disliked, The Coordinator fails because it is so explicit. Normally, many commitments made in the workplace are somewhat ambiguous, and minor deviations from a schedule may often be gracefully overlooked. But in the case of The Coordinator, when a schedule slipped even a small amount, the slippage and those responsible were announced to all affected. By stripping away the ambiguities in everyday communication, and putting the burden of keeping track of schedules and commitments on the system, the designers eliminated a very valuable form of flexibility from the work environment. The effect might be compared to a precocious child loudly pointing out the foibles and slips of guests at a dinner party. The Coordinator interferes with the ability to gracefully adapt to a change in circumstances.

The second problem with The Coordinator is that the first person to admit to a schedule slippage may well bear the brunt of the blame. Consider the following scenario: programmer number one reports that he'll need two more weeks to complete his project; the schedule is readjusted; three other programmers, who were also going to miss the deadline, breath (silent) sighs of relief. From management's viewpoint, programmer number one is solely responsible for the slippage. In contrast, without The Coordinator, it seems likely that the schedule slippage would have been noticed more gradually, and responsibility would have been more widely distributed. This is a subtle version of Grudin's 'unequal distribution of cost and benefits' problem–here the costs are not overhead for using The Coordinator, but are the consequences of its use (i.e. blame), and the inequality of distribution is an unnaturally sharp focusing of the blame.

A third negative effect of The Coordinator was that it deprived managers of an aspect of their power. In a large organization it is often difficult to find out what the real story is. Thus, there is status attached to knowing what is going on, when it will be done, and who is responsible. Because the actual validity of the schedule is known only to a few, there is a lot of status and power in being on top of things.

But since this information is only a few key presses away with The Coordinator, this whole power structure is undermined. Though this might be a positive result if it happened, the fact is that it's no result at all if it causes the organization to reject The Coordinator.

These shortcomings of The Coordinator are interesting. The Coordinator was produced by well-meaning and careful designers, and was based on a sophisticated analysis of how cooperative work is carried out. But their sophistication and care did not prevent them from designing a system that a substantial number of users hated. Note also that the problems with The Coordinator do not seem to be obvious to users–at least, before they use it. Most managers find a description of The Coordinator's features very appealing, but fail to anticipate the effect that The Coordinator has on the group as a whole.

REAL WORLD CASE STUDIES

In this section, I survey case studies of both normal cooperative work (i.e. not computer supported), as well as results obtained in experiments with CSCW systems. This differs from the previous section in that the papers described here make no pretense of generality. Each looks at particular problems (or particular behavior) relevant to CSCW observed in a specific case study. Though the researchers doubtless believe that their conclusions have validity beyond the particular case which they've studied, it's important to be aware that so many things can differ from one group to another that generalization is dangerous. The most important thing to notice in this section is the variety and complexity of the issues which can assume importance in CSCW.

When end users fail as designers[3]. This was one of a number of studies on the effects of introducing computer technology into the Scandinavian medical industry. One of the hallmarks of Scandinavian CSCW is the (legally required) inclusion of the actual end users of a system in the design of the system. In this study, nursing supervisors and researchers participated in the design of a system for supporting management-nursing supervisor cooperation in the nursing ward of a large Norwegian hospital.

The most interesting aspect of this study is that some of the nursing supervisors' initial recommendations turned out to be completely wrong. The specific example given was the problem of double filing: the filling out of multiple copies of reports, with one being kept by the nursing supervisor, and other copies being forwarded to appropriate managers. The nursing supervisors felt that one of goals of the system was to reduce duplicate work, and saw the elimination of double filing as an obvious step towards that goal. Thus, they were in favor of the plan to replace the current double filing system with a system where a single copy of the form was kept on-line.

But the nurses failed to recognize many of the benefits they got from double filing. Nursing supervisors used their personal copies of the forms as checklists, bookkeeping systems, and evidence that they had, in fact, done their part of a cooperative task. If questioned, they could out a marked-up copy of the form from their files. They also used the copies for taking notes to remind themselves of any oddities concerning the document, which may or may not be visible on the original. Another advantage of the paper forms was that it was easy for nursing supervisors to distinguish between approximate and accurate, or temporary and final reports: some reports were written in pencil, or annotated with question marks or comments in the margin. Most of these uses of the forms would be rendered impossible, or at least much more difficult, by the switch to on-line forms.

Although it can be argued that this is a problem of technology, since in the ideal case one should be able to scrawl annotations on even on-line documents with appropriate input devices and software, that's not the point. The point is that the nursing supervisors' intuitions failed. The nursing supervisors did not believe that they could scrawl notes on the electronic forms; they didn't think they'd have their own personal copies of the on-line forms. Rather, they simply weren't conscious of the role that little things, like annotating a paper copy of a form, played in their work.

The lesson here is that involving the end users in the design process is not a panacea. End users' intuitions fail, even when applied to how they do their own work. This observation, combined with Grudin's observation of the failure of designers' intuitions, emphasizes the difficulty of doing CSCW design.

Cultural resistance to changing the use of medical records.[4] This was another study on the effects of introducing computer technology into the Scandinavian medical industry. This study reports on difficulties encountered in attempting to change the use of the medical record from a 'personal tool' to a 'cooperative work tool.'

The background is that there are a number of large health centers which serve the population free of charge. A result

3. T. Bermann and K. Thoresen, Can Networks Make an Organization? *Proceedings of the Conference on Computer-Supported Cooperative Work*, September 26-28, 1988, Portland, Oregon, pages 153-166.

4. Y. Engstrom, R. Engstrom & M. D. Saarelma, Computerized Medical Records, Production Pressure, and Compartmentalization in the Work Activity of Health Center Physicians. *Ibid*, pages 65-83.

of this large scale health care is that a patient who returns to a medical center will often not see the same doctor as he or she saw six months previously. In this setting, computerized medical records become extremely important since they now need to serve as a communications medium between doctors about the ailments of the patient. This is in contrast to the traditional role of medical records, where one patient would normally see the same doctor, and the medical record was a much more personal and idiosyncratic tool.

The study focused on attempts to change the role of the medical record from a personal tool to a communicative device. It found that social factors were a much greater obstacle than had been anticipated. For example, physicians frequently failed to use the computerized medical records, or would glance at them in only the most cursory fashion. Subsequent interviews revealed that many physicians felt that seeking out and examining a colleague's chart for any other purpose than immediate care of a patient was a form of prying. Even in the group-oriented Scandinavian health centers, physicians perceived themselves as autonomous, and avoided actions–such as sharing medical records–which threatened that autonomy.

Another problem with the computerized medical records is that increasing the accessibility of the information may change the way it's used. That is, because computerized records are easier to obtain (in a database on a network) and easier to read (typed rather than scrawled), it is more likely that it will be used for supervisory or evaluative purposes. If physicians know that what they write down will be scrutinized by other doctors, managers, or even, possibly, used as ammunition in law suits, they may be reluctant to record what they really think, if that differs from what is 'safe.' It is ironic–though quite believable–that attempts to optimize the communicative potential of a tool may actually decrease its communicative value as a result of the political and legal realities of the workplace.

The moral. When you change the accessibility of information, you may change the way in which it is used. When you change the way it's used, you may change the nature of the information that is recorded. When you change what's recorded, it may no longer be what you wanted to make accessible.

Status and Communication: Who's in charge?
Anthropologists and sociologists have studied things like status hierarchies and authority negotiation for a long time. (Stripping away the jargon, this boils down to what's the pecking order and how do I show you where I stand in it.) More recently, CSCW researchers have begun to realize such concepts are important to them.

The Coordinator is an example of a system which ignores status hierarchies and authority negotiation to its detriment. One way it ignores the status hierarchy is that its description of a conversational 'action' doesn't take into account the participants. To The Coordinator, a request is a request is a request. But in the real world, a 'request' from the CEO is very different from a request from a subordinate. And in the real world, one way status is recognized and expressed is by changes in the way that something is communicated to a 'higher status' or 'lower status' employee. If The Coordinator eliminates an important way of signaling status, it could cause problems. While the studies to show whether this is the case have not been done, it is suggestive that The Coordinator is most accepted in traditional corporations where the status hierarchy is most clearly defined.

But just being aware of status hierarchies may not be enough. One potential pitfall for CSCW applications is that they may make assume that the status of group hierarchy is fixed. However, a paper by Charlotte Linde claimed that, in most organizations, status hierarchies shift from moment to moment, depending on task variables[5].

Police helicopter missions were studied because the work is extremely cooperative, and requires a lot of communication. Although the two members of the team–the pilot and the flight officer–are formally of equal rank, the pilot was actually of higher 'status' in the team, as determined by various metrics derived from anthropology and sociology. However, during the actual cooperative work, the authority status of the two participants was renegotiated from moment to moment, depending on whose task was most salient, who had the appropriate knowledge to deal with the current situation, and so on. In short, the person who was in control of the interaction varied depending on what they were doing, regardless of who was officially the boss.

Note that this authority negotiation was not a disruptive power struggle; rather, it was an unremarked part of the background of the ongoing task. It was generally reflected in the degree of indirectness of requests, with the person momentarily in charge issuing more direct requests than the other crew member. Thus 'Fly north' is more direct (and thus a way of taking charge or indicating control of the interaction) than 'Do you think we should fly north?' These findings have three implications for CSCW. First, CSCW systems must not assume that there is a fixed hierarchy in an group. A system which allows only one group member to control the proceedings may hinder cooperative work. Second, a CSCW application should allow for a changing hierarchy–that is, it should allow control of the interaction to shift from member to member. Finally, the ideal CSCW system should allow the shift of authority to

5. Charlotte Linde, Who's in Charge Here?: Cooperative Work and Authority Negotiation in Police Helicopter Missions. *Ibid.* pages 52-63.

be signaled in a natural (that is, subtle and unobtrusive) manner.

Conflict and Consensus in Group Decision Making. A certain amount of conflict is an inherent part of most cooperative work. In the ideal case, constructive conflict leads to more thorough consideration of a wider number of alternatives. Thus, one important issue designers of CSCW systems must be aware of is how their systems affect conflict within a group. An example of this is that communication carried out via electronic mail tends to be much less polite, and can generate more conflict than would occur in its absence.

One CSCW '88 paper reported research on the effects of a group decision making system on the way the group managed conflict, and how that, in turn, influenced the effectiveness of the group.[6] The researchers compared the effectiveness and amount of conflict in groups using the system, with groups doing the task manually. The CSCW system studied offered support only for the meeting process; it did not provide special communications facilities or allow the meeting to occur on-line. For the most part it simply automated tasks that would have been done anyway–for example, providing a place to list the various alternatives the group had come up with. But the designers included another feature: it allowed meeting participants to easily cast anonymous votes.

On the face of it, voting seems like a harmless feature to offer. After all, it could be done in a regular meeting, though making it anonymous is a little bit of trouble. And certainly, in most meetings, the opinions of the participants are often known. However, the research showed that the voting functionality was more than just a harmless frill. It was observed that the voting system was often used in a negative way– to cut off discussion of alternatives. The researchers found that groups which used the system tended to explore fewer alternatives, exhibit less change from their initial position, and achieved less consensus than groups which worked without computer support.

What's the moral here? It's not entirely clear. After all, the groups that weren't using the system could have used voting–anonymous or otherwise–in the same negative way. But they didn't. Perhaps it's a case of the if-you-have-a-hammer-everything-looks-like-a-nail syndrome: 'there's a special voting system, so let's use it.' The researchers suggested that training in how to use the system appropriately, or perhaps just greater familiarity, would eliminate the negative effect. That's quite possible. But the moral that I take away is this: even providing computer support for something that can be done easily without it can make a difference.

Shared workspaces. The popular image of full-blown CSCW system has someone sitting comfortably in front of a screen displaying an image of a remote coworker, easily chatting and gesticulating. A window, or perhaps a second screen, shows a common workspace where the data they are working with is displayed. Changes made by one participant are instantly visible to all.

But this may not be enough. Two studies of the ways in which designers used shared workspaces–sketch pads, whiteboards, blackboards–to collaborate, suggest that CSCW systems not only need to display the shared data, but also need to be able to show the gestures that coworkers make in relation to the data.[7] [8]

It was observed that drawings in a shared workspace often served as a sort of central information repository. Design often proceeded by one group member sketching an idea in the shared workspace, and then, as discussion proceeded, other members would add to or modify the sketch, until it evolved into a distinct drawing understood by all group members. It was also observed that designers often rapidly switched between drawing and writing, pausing to annotate part of a drawing, and then continuing with it. Often a transition from drawing to writing and back to drawing would occur in the course of a few seconds.

In addition to the drawings themselves, the gestures that occurred around the drawing surface were also quite important. The importance of gesture in design is easy to illustrate. Suppose you have an audio tape of a collaborative design session, as well as the drawings and notes generated in the session. If you just listen to the tape, you'll find that the words may carry very little meaning. 'Move *this* here, but do *that* to it', is a very typical remark. Second, if you examine the drawings generated in the meeting, they'll often appear to be doodling or meaningless scrawls. Even listening to a tape while looking at the final drawings doesn't capture the full content of the meeting. When designers work together, gestures themselves convey meaning. It is gesture that binds the words and drawing into a meaningful structure.

A third observation was that the gestures which occurred around the drawing surface also functioned as a mechanism for controlling the group's interaction. Sometimes gestures were used to focus the group's attention on a particular

6. M. S. Poole, M. Homes and G. DeSanctis, Conflict Management and Group Decision Support Systems. *Ibid.* pages 227-243.

7. J. C. Tang and L. J. Leifer, A Framework for Understanding the Workspace Activity of Design Teams. *Ibid,* pages 244-249.

8. S. A. Bly, A Use of Drawing Surfaces in Different Collaborative Settings. *Ibid,* pages 250-256.

drawing. At other times gestures were used as a means of controlling the flow of conversation. A gesture could be used to indicate the desire to take a turn speaking, or to elicit the input of another group member. It was observed that in CSCW systems which did not provide a shared workspace that supported such attention-getting gesturing, interaction was greatly reduced.

These two studies suggest three features of CSCW applications. First, there needs to be a common workspace on which all members can draw. Second, the software should support rapid transitions between drawing and writing. Third, not only should the workspace support drawing, but it should also be able to depict gestures made by the various participants at, and in the vicinity of, the drawings. As yet, there are no CSCW systems which offer this sort of support.

Real World Case Studies: Summary. This section has described a wide range of factors which are important in the design of CSCW systems. Some of these factors are things which are missing from current CSCW systems–for example, support for gestures in relation to the shared data. Some of these factors are things that exist in the everyday world–taboos against 'prying'–but go unnoticed until a CSCW system tries to violate them. Sometimes the introduction of seemingly attractive functionality has unforeseen consequences, such as The Coordinator's elimination of ambiguity. And sometimes simply providing computer support for something that could be done without it has negative consequences, as with computer support for voting during a meeting.

These are all problems that CSCW designers have to deal with. Any one of them could cause the failure of a commercial CSCW system. And yet, it's not clear how to avoid them. As the first study we looked at showed, even including the actual end users of a system in the design process may not help avoid such problems. What can be done? One direction in which some hope can be found is in the inclusion of 'outsiders' in the CSCW design process. By outsiders, I mean professions other the computer science, cognitive psychology, and graphics design (even though the latter two are relative newcomers to software design). The next section of this article, which describes laboratory prototype work, includes some good examples of using 'outsider' expertise to understand and solve some difficult problems.

LABORATORY PROTOTYPES

This section describes two prototype systems which are in the process of being designed. Cruiser is at an extremely early stage of design–it is primarily a video prototype. The Capture Lab is more fully realized, but it is still evolving, even as it is used as a vehicle to study the meeting process.

Cruiser[9]. Most CSCW researchers focus their efforts on supporting cooperative *work*. Robert Root, of Bell Communications Research, described a different approach to CSCW. Root argues that informal social processes and unplanned meetings and collaborative efforts are at the heart of most cooperative work activities. (Those who question the importance of informal social interaction in the workplace will find a pointers to a large body of relevant research in this article.) Thus, supporting cooperative work requires tools which support the formation and maintenance of informal social relationships and unplanned interactions.

Root describes a system called "Cruiser" which is based on a "social browsing" metaphor. Just as a person may browse in a book store, looking for some undefined but interesting book, so people may go out looking for unplanned social interactions. For example, someone may wander down a hallway of coworkers, glancing in the offices to see if anyone is around. Another example is cruising, the stereotypic teenage activity of the 50's, for which the system is named.

Cruiser provides a set of virtual hallways and a set of virtual offices (i.e. video windows into real offices) for each hallway. As in the real world, virtual hallways afford a means of getting to the offices of coworkers. But, unlike the real world, Cruiser allows users to define what offices the hallways connect, or allows the system to generate a hallway, either randomly, or according to some set of real world criteria (eg. common interests of occupants, physical location...). Finally, Cruiser allows users to either wander a hallway, glancing into its offices to see if someone is there, or to jump directly to a particular office.

Cruiser attempts to preserve the social nuances of the workplace. Because unwanted interruptions can be distracting and may have negative effects on social relationships, a great deal of attention was focused on interface features for indicating the availability of the occupant for social interactions. As might be expected, the visual and audio channels of the video window could be turned on or off separately. The state of these channels–by analogy to the real world–were always symmetric; that is, the office occupant could hear a visitor only if the visitor could hear the occupant. The visual channel could also be partially shut down; this was portrayed by a graphical image ("blinds") overlaid on the video widow. In addition, Cruiser provides doorbells for requesting an interaction (when visual and audio channels are turned off), as well as a mechanism (posted notes) for leaving notes on a door. Finally, posted notes may also be used to provide an automatic "return visit" function: when the occupant returns to the office and finds a note on the 'video window', mousing on the note will

9. R. W. Root, Design of a Multi-Media Vehicle for Social Browsing. *Ibid*, 25-38.

jump the occupant directly to the office of the person who left it.

Cruiser is not yet a real system. However, it is important for two reasons. First, it exemplifies an innovative approach to designing CSCW tools. Rather than trying to facilitate a particular task, or to solve the entire CSCW problem, Root identifies an important component of cooperative work and designs a tool to facilitate that. Second, Cruiser demonstrates that important perspectives can be provided by disciplines not usually associated with human interface design (social psychology; anthropology; sociology).

The Capture Lab[10]. A paper by Marilyn Mantei described design decisions made during the development of the Capture Lab, a computer supported meeting environment. Unlike many CSCW projects, the design goal was to build a *room*, not just a software environment. With this approach, the interface becomes a room interface, and consideration is given to the positioning of computers and people within the room, and the visual appearance of the room.

Before discussing the design decisions which are the foci of the paper, note that the designers of the Capture Lab approached the implementation of a CSCW system with unusual care and caution. To quote the paper: "Our decisions were made with the goal of making the Capture Lab meeting environment as close as possible to the meeting environments that our user population was accustomed to. We adopted this goal because we did not know what impact our computer supported meeting environment would have on groups."

In brief, the capture lab is a room containing a large "electronic bulletin board" at the front, a table with Macintosh II's inset into the surface, and software permitting users to transfer information from the screens of their Macintoshes to the electronic bulletin board. The paper focused on the observations and rationales behind design decisions involving seating arrangements, inter-viewing distances, and interior decoration of the room.

The initial seating layout of the Capture Lab consisted of two semi-circular rows of wedge shaped desks facing a front screen. However, it was observed that this seating arrangement resulted in much less interaction than when group members sat around an oval table. Mantei speculated that a reason for the decreased interaction was that a semi-circular seating arrangement was not conducive to eye contact, facial expressions, gestures, or other nonverbal exchanges or signals. The redesign of the seating layout involved sinking Mac II monitors into the surface of an oval conference table (so as not to impede eye contact), and using swivel and roll chairs, so that meeting members could easily turn to face the electronic blackboard at the front of the room, and then swivel around to face the meeting participants.

In a regular conference room, the manager tends to sit at the front of the meeting room, opposite the door. This is known as the 'power seat'. From the power seat, the manager can face the people at the meeting, monitor those who come and go, and easily use the whiteboard or overhead projector. However, in the Capture Lab, the location of the power seat has changed. The content of the electronic blackboard is controlled from individual keyboards scattered around the table. The brightness of the electronic blackboard makes viewing from the front seats more difficult. A manager sitting in the supposed 'power seat' has the feeling of being in a visual shadow, since something being typed onto the screen draws participants' attention away from the manager.

In the Capture Lab, the real power seat is at the back of the room, opposite the door. This location allows the manager to easily watch either the screen or other participants. It also causes other participants to turn completely away from the screen when the manager speaks, thus minimizing the chance of distraction.

Managers who have used Capture Lab have been observed to initially sit in the traditional power seat. However, they soon begin to exhibit signs of discomfort; eventually (usually during the first meeting) they move to the new location of the power seat. In addition, perhaps because it is so easy for anyone to enter information onto to electronic blackboard, the managers were observed to give up using the keyboard. Instead, they adopted a style of either verbally directing someone to type in their comments, or borrowing someone else's keyboard when they wanted to enter something. Both types of behavior tend to emphasize and enhance the manager's control over the meeting. An interesting incidental observation was that participants would form different seating arrangements in meetings in which a person in a higher managerial position was present. In such cases, participants would move their chairs away from the table and form a semi-circle; this phenomenon was even observed in mid-meeting, when a high level manager showed up late.

As already noted, it is important for meeting participants to see one another so that non-verbal exchanges and signals can facilitate verbal exchange. As it turns out, not only must participants be able to see one another, but they must be sufficiently close to one another. One consequence of embedding the monitors in the conference table to enable

10. M. Mantei, Capturing the Capture Lab Concepts: A Case Study in the Design of Computer Supported Meeting Environments. *Ibid.* pages 257-270.

eye contact, was that the conference table had to be wider than normal. This distance was sufficient to decrease verbal interaction (people tended only to speak to those next to them). Because the physical size of the monitors prevented any reduction of the actual size of the table, the designers of the Capture Lab resorted to a redesign of the room's interior suggested by an architectural consultant. The redesign involved the use of contrasting colors in the table, carpeting, and wall coloring, and was intended to reduce the apparent visual distance, giving the individuals a sense of being closer together. The use of color to shorten apparent distance has worked; participants now talk to all other members of the meeting, in spite of the size of the table.

The basic design goal of the Capture Lab was to change the meeting environment as little as possible. This cautious approach was well-founded. As Mantei notes in her conclusion, even the few alterations that were made to the standard meeting environment produced dramatic changes in the behavior of meeting participants. In Mantei's view, the most important result of her research is not the particular observations that are reported–since these may vary depending on the group, task, and corporate culture involved–but rather the more general observation that "small perturbations in the design of a meeting environment can interact with the meeting process." This is something that I think most advocates of groupware fail to recognize; even those in the know often fail to grasp what large and unexpected effects may result from seemingly insignificant changes. Just think, if Mantei hadn't thought of bringing in an architect to reduce perceived visual distance, or had simply neglected to get swivel chairs with rollers on them, the Capture Lab might have failed as a CSCW environment.

CONCLUSIONS

It is easy, with twenty-twenty hindsight, to condemn the design of a CSCW system as being badly done. However, the predominance of unsuccessful groupware applications suggests that the problem is not bad designers, but rather that the design of groupware is extremely difficult.

This article has talked about a number of rather weird things. The role of table width in the interactivity of a conversation. Seating arrangements and the importance of having rollers on chairs. Using interior decorating tricks to reduce perceived distance. The importance of ambiguity in communication. Why changing the accessibility of information may change the nature of the information. Status hierarchies and the importance of allowing subtle and unobtrusive means of authority negotiation. Inequalities in cost/benefit distribution across a group.

These aren't your standard human interface concerns. But, as increasing connectivity becomes more and more predominant, and 'group friendliness' becomes more and more important, such topics will have to be understood and addressed. Anthropologists, social psychologists, sociologists, architects, and even interior decorators may all have their place on the design teams of the future.

Caveat. One of my goals has been to paint a coherent picture of the CSCW field. To the extent I have succeeded, the reader should beware. The field of CSCW is not at all coherent. Journals are filled with conflicting design approaches, inconsistent research results, and overly-optimistic solutions.

I have emphasized problems, rather than solutions or even research results, because the problems are the only things that I really believe in yet. Research results are subject to continual reinterpretation as new evidence rolls in. This year's crop of solutions may look less promising when implemented and tossed into the marketplace. Finally, with the forthcoming flood of self-proclaimed groupware, and the optimistic proclamations which will accompany it, it is important to emphasize the breadth and difficulty of problems which will also be in attendance.