Awareness and Coordination in Shared Workspaces

Paul Dourish and Victoria Bellotti

Rank Xerox EuroPARC
61 Regent St
Cambridge CB2 1AB UK
dourish@europarc.xerox.com, bellotti@europarc.xerox.com

ABSTRACT

Awareness of individual and group activities is critical to successful collaboration and is commonly supported in CSCW systems by active, information generation mechanisms separate from the shared workspace. These mechanisms penalise information providers, presuppose relevance to the recipient, and make access difficult. We discuss a study of shared editor use which suggests that awareness information provided and exploited passively through the shared workspace, allows users to move smoothly between close and loose collaboration, and to assign and coordinate work dynamically. Passive awareness mechanisms promise effective support for collaboration requiring this sort of behaviour, whilst avoiding problems with active approaches.

KEYWORDS: Awareness, information sharing, coordination, shared workspaces, shared feedback.

1 INTRODUCTION

Studies of collaborative writing [*e.g.* 2, 7, 16] highlight the extent to which information sharing, knowledge of group and individual activity, and coordination are central to successful collaboration. These factors are clearly critical concerns in the design of computer systems to support collaborative writing, and mechanisms for their support are the subject of this paper.

Information relating to these factors contributes towards what we refer to as *awareness.* In these terms, awareness is an *understanding of the activities of others*, which provides a *context for your own activity*. This context is used to ensure that individual contributions are relevant to the group's activity as a whole, and to evaluate individual actions with respect to group goals and progress. The information, then, allows groups to manage the process of collaborative working.

It is particularly important to recognise that the context within which group members collaborate is comprised of both the object of that collaboration and the way in which the object is produced. We must therefore consider as context not just the *content* of individual contributions, but also their *character*; their significance with respect to the whole group and its goals. It is only by providing awareness of both aspects of group members' work that systems enable each individual to make sense of others' activity and tailor their own work accordingly.

Awareness information is always required to coordinate group activities, whatever the task domain. Although we deal largely here with collaborative text editing, other collaborative activities can benefit equally from the approach we outline.

Systems described in the research literature take various approaches to the provision of awareness information. A primary distinction between these mechanisms is whether the information is *explicitly generated*, directed and separate from the shared work object; or *passively collected and distributed*, and presented in the same shared work space as the object of collaboration. The latter mechanisms are frequently restricted to synchronous systems (*i.e.* those in which all collaborators are virtually co-present and working at the same time); however, there is no need for this restriction and later we will discuss their value in what we term *semi-synchronous* systems, which incorporate synchronous and asynchronous working.

In the following sections, we describe in more detail some approaches to the provision of awareness information in particular systems, and we highlight a number of problems with these approaches. We then discuss a case study of groups performing an open-ended design problem using a particular shared text editor which embodies an alternative approach to the provision of awareness information. We show how the groups used this alternative approach, point to the ways in which it overcomes some of the problems in the more traditional approaches, and the ways in which it supports important aspects of awareness information for collaborative working.

2 AWARENESS IN COLLABORATIVE WRITING SYSTEMS

We will begin our discussion by looking at three particular collaborative writing systems, and the mechanisms they provide for sharing awareness information between collaborators. The designers of these systems explicitly acknowledge the importance of issues such as information sharing and coordination, and mechanisms for their support, but the approaches they take result in problems for groups, which we shall discuss.

2.1 Quilt

Quilt [6, 11] is an asynchronous collaborative authoring system developed at Bellcore. Rather than managing all aspects of document production, Quilt essentially forms a superstructure which manages the specifically collaborative aspects of group authoring; for instance, Quilt makes no assumptions about the underlying editors which might be used to enter and modify text. The Quilt developers identify the primary problems in collaborative authoring as *coordination*, which includes ensuring that work progresses and that redundant work is minimised, and *information sharing*, which includes both information about the substance of the work, the management of the work, and the interpersonal relationships within the group.

Quilt provides a range of explicit facilities to help support these aspects of the collaborative authoring process. The primary mechanisms available to Quilt users are a hypermedia system representing the document along with annotations, audit trail recording, and integrated electronic mail and conferencing systems. Annotation provides a mechanism by which users can comment to each other about the document material. Audit trail recording allows collaborators to review each other’s activities. Electronic mail and conferencing provide mechanisms by which users can discuss document-related and activity-related issues, and distribute information about current or planned activities.

Underlying these mechanisms is a system of configurable role-assignment, which controls the degree of access which individuals have to the document. The use of explicit roles provides information about the character of a participant’s activity; if you know that a colleague is a “reviewer”, then your uncertainty about her potential activities is reduced

Quilt, then, takes a very active approach to the challenges of information sharing and coordination in asynchronous collaborative authoring. The distribution of awareness information is an explicit activity on the part of each member of the group. Any uncertainty about the character of others’ work is reduced by imposing roles on all participants. Quilt manages roles through formal mechanisms, which means that explicit actions must be taken to change them.

2.2 PREP

PREP [14] is an asynchronous “work in preparation” editor which can be used by groups to collaboratively author documents. It concentrates in particular on the early stages of the writing process; idea generation and collection, initial text production, commenting and revision. The structure of information in PREP is very general. Information “chunks” can be linked together to form “drafts”, and to form matrices which relate parallel information streams. The interface simplifies this model by displaying the information in a column structure, presenting the information streams in parallel and using spatial layout to emphasise the linkages between them. A typical view, for instance, might present four such columns; one to hold a plan structure for the document, one to hold draft text, and two for the annotations made by two collaborators.

Like Quilt, PREP uses role assignment to structure and delimit areas of responsibility and to control access to the information streams. The developers, however, acknowledge that, in natural collaboration, roles are typically fluid and continually re-negotiated, and warn of the danger of “premature” definition of roles in collaborative activity. Similarly, they point out that a direct mapping from roles to edit activities may be inappropriate; a reviewer may wish to be able to edit the text being reviewed, rather than merely make annotations, if that suits her particular work style. As a result, PREP tries to support a model of “communication about comments”, such that comments, annotation and linkages which collaborators may make are not their only means for expressing ideas relating to the text in preparation. Even so, collaborators are still restricted in their roles, and consequently in the activities which they can undertake at any time.

So, like Quilt, PREP uses roles, explicit annotation and structured or directed messaging to provide means for generating awareness and coordination information. Information about the content of activities in progress is divorced from the activities themselves, but must be provided separately through some other channel.

2.3 GROVE

GROVE [4] is a synchronous, multi-user editor for the creation and editing of textual outline documents (tree-structured documents which may be viewed at various levels of specificity). It is designed for use in both face-to-face and remote collaborative situations. While GROVE takes advantage of audio communication to support informal awareness between participants, the system's representations of other users' activities are implicit. A GROVE window will show other users' text entry for any outline nodes which are also opened on the local user's screen, and each node is marked as open, closed or terminal (*i.e.* without any sub-nodes). Parts of the document are presented to each user through a *view*, which may be public, private or shared. In addition, access control mechanisms can be used at each point in the tree to control who can see, edit or create a node.

GROVE differs from Quit and PREP in a number of significant ways. It has no explicit notion of role, although the use of access control does correspond to some of the (non-awareness) aspects of role use in collaborative systems. The notion of "view" serves to differentiate the information presented to each user, which reduces the extent to which the shared document can act as a common resource for reference. Although the dOPT synchronisation algorithm [5] which underlies GROVE can manage simple text streams as well as outline documents, GROVE's structuring of the document also implicitly serves, in some ways, to structure the activities in which the group engages. So, while GROVE is based on a model of synchronous collaboration and external communication channels, it constrains some task elements in an effort to prevent editing conflicts and provide a mutual awareness of activities.

All of these systems clearly embody an assumption that a simple awareness of other's activity needs to be augmented with other explicit, or restrictive mechanisms for ensuring an easy collaboration, such as annotations, role assignment, access rights and so forth. In the next section, we begin to question such assumptions by considering a number of problems implicit in these awareness support mechanisms.

3 MECHANISMS FOR AWARENESS INFORMATION

The previous section briefly described three collaborative editing systems, with an emphasis on the mechanisms they use to support sharing of awareness information between participants. In this section, we describe two general models which are used, and some of the problems which they generate.

Existing CSCW systems vary in the mechanisms they provide to support awareness. One mechanism, which we refer to as *informational*, is to provide explicit facilities through which collaborators inform each other of their activities. For instance, software control systems such as RCS [18] ask users to provide text for an "edit log" which describes the nature of changes; or electronic mail can be integrated with an authoring system as a channel for sharing this information, as in Quilt. A second mechanism, which we refer to as *role restrictive*, arises from explicit support for roles in collaborative systems. A role describes an individual's relationships to the shared work objects and to other participants, and is typically linked to a set of operations which can be performed. In a shared authoring system, for instance, an "author" role might be associated with the read, write, create, delete and edit operations, while a "reviewer" might be limited to read and annotate. One of the effects of explicit role support, then, is to reduce uncertainty about the actions an individual might take, and hence provide greater awareness amongst participants of each others' likely activities. However, awareness through roles provides information only about the *character* of the activity, not the *content*.

3.1 Problems with Informational and Role-Restrictive Approaches to Awareness

While both the informational and role restrictive approaches are useful in conveying to collaborators an awareness of progress and joint activity, they also have some problems. We identify three in particular.

Firstly, the user who provides the information does not directly benefit. In the role restrictive case, we encounter a significant trade-off with respect to benefits. The price of heightened awareness for the group is clearly *restriction in the potential activities* of individuals. However, there is a further problem for role restrictive CSCW systems. This is that, although explicit roles may allow for easier social organisation of collaborative activity in conventional interactions and collaborations, one often observes roles being negotiated and reassigned dynamically. This phenomenon has been identified in other computer supported meeting situations where participants are released from the tyranny of restricted access to shared work spaces [*e.g.* 1, 13]. There seems to be justification for arguing that role-switching in CSCW systems should, therefore, not be a complex, time consuming operation which hampers this negotiated process.

In informational systems supporting awareness, the individual is required to supply the information; again, this cost is repaid to the group rather than the individual, and adds an extra work load in the case of computer support over natural collaborative work. The problem of correctly matching benefits to the individuals who incur costs is one of the problems cited in Grudin's analysis of the failures of some collaborative systems [9].

A second problem with these approaches to awareness can be seen as a proviso to our previous statement that other individuals benefit from the action of one group member in reporting activity. The others *may* benefit, but this is by no means guaranteed. Individuals will receive *what the initiator of the information deems to be appropriate*. However, appropriateness can only be determined in the context of the other individuals' activities. So, information provided in this way may not be at the appropriate level of specificity for the receiving individuals, or it may not be relevant to their particular activities at the time, or it may reflect different assumptions about aspects of the joint work.

The final problem concerns the way in which the information is made available by informational mechanisms such that *delivery is controlled more by the sender* than by the recipient. However, the sender cannot predict what information will be needed and when. The information is not continually available to be browsed and particularly relevant information cannot necessarily be separated from other information which is less relevant to the recipient's particular activity at that moment. Thus, not only is the producer of the information burdened in the act of producing, but the recipient is restricted in ways of using the information.

4 SHARED FEEDBACK

In contrast to these approaches, we present a case study of a collaborative text preparation tool which embodies some (but not all) aspects of what we call the *shared feedback* approach. Shared feedback makes information about individual activities apparent to other participants by presenting feedback on operations within the shared, rather than the private, workspace. We will describe the particular system under examination, and then proceed to show the way in which users exploited the shared feedback features in the completion of their task.

4.1 A Case Study—ShrEdit

We collaborated with Judy and Gary Olson in a study of groups of designers using an application called ShrEdit [12] whilst solving design problems. ShrEdit is a synchronous, multi-user text editor which runs on a network of Apple Macintoshes. It was developed as a tool to explore the support of design meetings. Before discussing observations from the study we will describe some of ShrEdit's features.

ShrEdit allows multiple users to edit a set of documents collaboratively. Each user can have a number of shared and private windows. A shared window presents a view onto a shared document; each user has an edit cursor within each shared window, allowing them to edit text concurrently. Views of documents are unique to each user; each user's window can be differently sized, and aligned on a different part of the document. Private windows contain documents which only one user can see and edit, and can be used for making notes or creating text which may later be pasted into a shared document.

ShrEdit locks shared windows at the level of text selections. No user can edit text which has been selected by another user. Similarly, it is not possible for two users to have their edit cursors at the same point in the document. ShrEdit does not have telepointers which would allow individuals' mouse-cursor movements to be seen by other collaborators, and other users' edit cursors are not displayed to an individual. However, other users' edit actions are displayed in all shared windows with a low latency, and cursor "collisions" are indicated with an audio signal and a pop-up window.

A control window associated with each edit window displays the names of the participants in the session. Through this window, users can "find" other users; ShrEdit scrolls to the current location of their edit cursor in the document in the window. They can also "track" others, which means that they see another user's view (as far as possible given differences in window shape) complete with the edit cursor and text selections; this persists until switched off. Each control window records whether participants have a selection in the associated window, are tracking someone else, or tracking you. Figure 1 shows a typical layout of public, private and control windows.

ShrEdit avoids imposing a structure upon users' activities. There is no strong model of the collaborative editing *process* behind its design. All participants have equal access to the shared document windows and can type at any time. Nor does ShrEdit provide sophisticated functionality to support awareness beyond showing everyone's text as it is input, and giving rudimentary information about whether participants are editing, or are tracking someone. This freedom allows users to adopt very different working styles.

4.2 Description of Study

Groups of three designers (all with previous experience of working together) tackled open-ended design problems using ShrEdit. Since one of the aims of the study was to simulate remote collaboration, we placed them in separate locations, linked via video and/or audio. The informal channel of verbal communication, whether across a meeting room table or via audio, is important for supporting a system of this flexibility.

After a training period, each group was given two 20 minute practice problems and finally a 90 minute design problem to be collaboratively solved using ShrEdit. The final problem was to design a 24-hour unstaffed "automatic" post office offering a subset of the usual post office services, such as selling stamps and weighing parcels. The designers were asked to write a plan for the services their design would provide, how it would work, and things they would need to investigate further, as well as to make notes on their reasoning. The design problem was carefully worded so as to

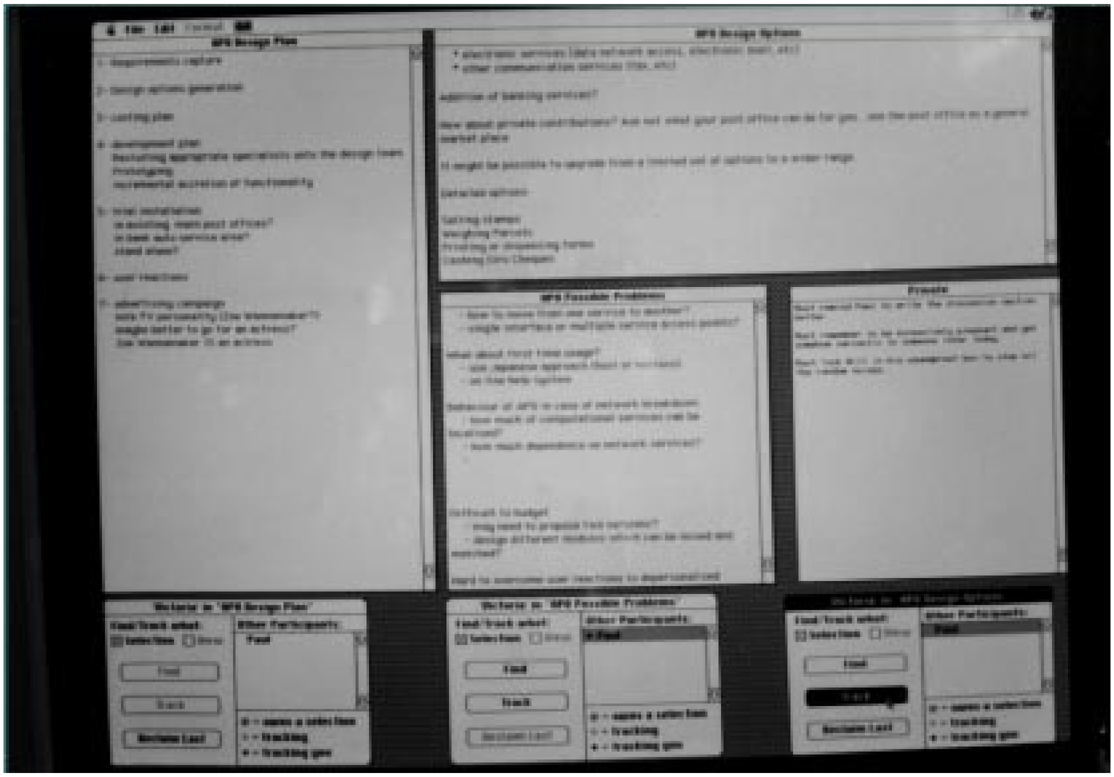

FIGURE 1. A ShrEdit screen, showing three public windows, one private window (middle right) and, at the bottom, three control windows.The control windows show other active users and provide “find” and “track” facilities.

avoid the need for drawing, which ShrEdit cannot support. The experiment ended with a debriefing session where the designers were interviewed about their use of ShrEdit, criticising and suggesting improvements to it. All stages of the study were video-recorded for later analysis. For a fuller description of the study see [15].

Designer groups were free to choose the number of shared windows they wanted, how they would use these, and the way they would work together. We were struck by the diversity, not just of the design solutions generated using ShrEdit, but also of the ways in which groups produced them, which they later commented were compatible with the way they would normally work together.

In the next section we highlight some ways in which awareness information was critical to ShrEdit users, and some examples of how its absence was problematic. We base our report on analysis of video-tapes of four groups of users and restrict our examples to the final experimental problem.

4.3 Some Observations on Use

The shared workspace provided a focus for the designers’ work and discussions. Talk dominated the activity of the designers with many periods when nobody in a group was typing, whilst two or all three talked. Even when everyone was typing, there were frequent sporadic or more sustained bursts of conversation. Subtle inflections of voice, interruptions, humour, or just grunts were also used to convey or cut off information where appropriate. Not only was speech allowing exchange of design ideas, it was also used to maintain awareness of group members’ activities with a very low overhead in terms of effort required.

This talk was very much contextualised by, and related to, the shared grounding which the synchronous shared editing allowed. Much of the conversation referred to or implied a shared context provided by the shared documents. We also observed many instances where the information being generated in the shared workspace, and the way it was being organised, acted as a focus that tended to curb digressions and to keep the group working in a coordinated fashion. So awareness of others’ work through ShrEdit enabled group members to organise their activities and provided impetus for design contributions.

Participants continually moved between concurrent, but more or less independent, work, through discussions and coordination, to very tightly focused group consideration of single items. These movements were opportunistic and unpredictable, relying on awareness of the state of the rest of the group.

The activities of the group also varied continuously, comprising permutations of individuals typing, editing (*i.e.*

correcting spelling, indenting and so on), reading, talking, listening or thinking. For example, we frequently observed that one designer would stop typing and watch the contributions of one or both of the others. This might prompt further debate and reorganisation of the document or participants' roles. Sometimes two designers would stop and talk whilst the third might ignore them and continue working independently or suddenly chip in. Occasionally one participant would act as a scribe during discussions, but here, of course, the record could be seen and modified by others.

Designers were clearly aware of the special status of authorship and tended to partition responsibility for and rights over different parts of shared documents. They would ask who had written certain things or warn each other about changes which affected others' work. In the example below, the group has adopted a convention under which each user has their own window, which nobody else should write in but everyone can read.

D3: Um guys, maybe we should copy the stuff from window three, which seems to be the heading, into the main window and then start doing our own thing in our own window.

D2: Into our own, OK.

D1: Do you want to do that [D3] seeing as it's your stuff?

The lack of structure in ShrEdit removed work-process constraints, and continuous awareness allowed users to vary their activities dynamically and opportunistically in response to the changing state of affairs with the group and the growing document.

Some problems with ShrEdit concerned the extent to which users could observe others' activities. There were two facets to this; seeing the *character* of another's activity, and seeing the *content* of another's activity. Often users would watch each other typing, or ask what they were doing. Surprisingly, not much use was made of the "find" and "track" facilities. In debriefing sessions users claimed that this was due to clumsiness in the control window interface. Users preferred to ask where others were, or just scroll to where they knew they would be. One group developed an indexing and indentation scheme so they could give accurate location references. This scheme also gave implicit information about the character of the work being done by participants.

There were also problems with informing others about what you were doing. Users often volunteered such information to the group, as below:

Two designers are working together on part of the document whilst the third is attending to another part. The third designer alerts the other two to a change, as opposed to an addition, he wants to make.

D1: Lets make the first, designer stamps from preset selection.

D2: OK... Now I'll copy this; I'll cut this...

D1: Yeah cut that stuff below and put it in phase three.

D2: ...I can't cut that, I'll just copy that down to...

D3: I don't think there's "no salaries to pay", it's "fewer". You've got to have some kind of fix it.

D1: Huh?

D1: What are you doing [D2]?

D2: What? ... I'm doing... I'm down in the fax stuff.

Having others in the group be aware of what one was doing seemed to be extremely important. Very little use was made of private windows, although this could be due to pressure on group members to produce as much joint work as possible during the experiment. Designers frequently described what they were doing and sometimes would explicitly ask others to look at their work. One group, mentioned earlier, assigned each participant a shared window into which, by consensus alone, only they could type. They also had a window into which all three could type. In this way everyone could see each other's individual work, and, furthermore, the status of that work although public, was clearly an individual's contribution rather than the group's offering.

During debriefing, the users repeatedly emphasised the importance of awareness of work activities, by making design suggestions for easier access to this information. At the same time the designers were very positive about the freedom that ShrEdit gave them to work in a manner which suited them.

4.4 Use of Shared Feedback in ShrEdit

ShrEdit provides some of the functionality of the shared feedback approach, in that it automatically represents activity within the shared space. It does so without any explicit informational and role restrictive mechanisms to facilitate collaboration, such as role assignment, setting access rights and so forth. In spite of the lack of such mechanisms, its users are still amply aware of each other's activity, such that they can negotiate and adapt the content and character of their own work with respect to the context of group activity, and can organise the group's activity in a flexible but coordinated manner. In other words, they succeed in organising their collaboration without requiring the effort of explicit, system-structured exchanges of information about their activity, or the restriction of their contribution to some predetermined, inflexible role. They do so in a manner which is subtle and dynamic rather than formal and static. There is clearly much to be learnt about the way people collaborate before we can presume to preordain how that collaboration should be structured.

It should be pointed out that, although ShrEdit embodies aspects of the shared feedback approach, it still lacks certain other features which we would see as important; indeed, in providing some features of shared activity awareness, ShrEdit highlights the lack of some others. For example, whilst users could see other's input, they could not see their edit cursors. It is not surprising, then, that they complained about this during the debriefing.

5 DISCUSSION

“Awareness”, principally of participants’ activities with respect to a collaborative context, is a critical issue for collaborative systems and one to which the developers of Quilt, PREP and Grove explicitly address themselves. It is fundamental to coordination of activities and sharing of information, which, in turn, are critical to successful collaboration. Awareness plays a number of key roles. First, high-level awareness of the *character* of other’s actions allows participants to structure their activities and avoid duplication of work. Second, lower-level awareness of the *content* of others’ actions allows fine-grained shared working and synergistic group behaviour which needs to be supported by collaborative applications.

We have already discussed problems which arise from the approaches to awareness in some existing CSCW systems. We have also shown how a system which embodies aspects of the *shared feedback* approach was used by groups to flexibly coordinate their activities. We now consider some aspects of shared feedback more generally, and show how it relates to the problems with other approaches which we identified earlier.

5.1 Shared Feedback: An Alternative Approach

An alternative approach to increasing awareness, which we have successfully used in other collaborative systems, is to automate collection and distribution of information, and to present it as background information within a shared space. This is the *shared feedback* approach; presenting feedback on individual users’ activities *within the shared workspace*. The emphases of this approach are on low overheads for the providers and recipients of awareness information, availability of information as and when needed as a context for individual activities, and avoidance of restrictive pre-structuring of group activity. This approach is commonly associated with exclusively synchronous applications, although this is not, in fact, a requirement. The notion of semi-synchronous, persistent shared workspaces leads to non-synchronous shared feedback, and we shall return to this presently.

What are the particular benefits of this workspace-based shared feedback approach? The ShrEdit study begins to suggest some of these. We have observed how individuals have the opportunity to peripherally monitor others’ activities, and comment on them, so that an individual, even when working independently, is both communicating her activities (allowing others to avoid duplicating her work) and providing others with the opportunity to comment on the activity or observe consequences for their own actions. This is achieved without increasing the workload of the individual “producing” the information. Conversely, users can explicitly tailor their contributions knowing that others can see them, so as to convey information and solicit responses, via the shared workspace or some other communication channel. At the same time, by reducing the need for role restriction, this approach is compatible with a more flexible model of role-assignment which supports the fluid negotiation and reassignment of role we see in a number of collaborative activities. These mechanisms for coordination and information sharing can be made available in a collaborative tool orthogonally to the task itself; they can be applied to a range of tools which embody particular work styles, or to a single tool which can be used in multiple ways. This approach is the one taken by ShrEdit and allows for the adoption of different working styles and self-organisation of the collaborative process which we remarked on in discussing our study.

We should not find it surprising that these features (the ability to passively monitor other’s actions, and to tailor indirect productions for other individuals who can see/ receive them) are useful ways of coordinating shared activities like collaborative writing, drawing or programming—they are mechanisms which we see groups use to coordinate natural collaboration in other settings. For instance, Heath and Luff [10] show how precisely these mechanisms are used to coordinate activities between individuals working together in the control rooms of a major urban transportation system. The same mechanisms also underpin the role of shared awareness in other contexts, such as the support for informal interactions within distributed work groups [3]. Tatar *et al* [17] point to a clear need in Cognoter, a group brainstorming support tool, for co-reference; feedback enabling users to see others’ work and actions as they occur, which allows them to communicate, interpret and coordinate their activities more efficiently. In addition to verbal and visual information, non-speech audio information can also provide a means for shared feedback in a variety of environments [8].

5.2 Semi-Synchronous Systems

In looking at different approaches to providing awareness of individual and group activities in shared workspace systems, a correlation emerges between synchronous collaboration and passive, workspace-based group feedback. It is worth considering whether this distinction in awareness mechanisms is intrinsic in the choice of synchronous versus asynchronous approaches. There are two related questions here: is it possible to imagine workspace-based awareness mechanisms in asynchronous systems, and is the distinction between synchronous and asynchronous approaches necessarily such a strong one?

Tackling the first question, we can certainly imagine asynchronous awareness information presented in the same workspace as the work object. A primitive example of this type of asynchronous awareness information is the use of "change bars"; margin marks indicating text areas which have been changed. Change bars are examples of document-based representations of activity. There is no reason why change bars or similar representations[1] cannot provide further information, such as the nature of changes, the identity of the collaborator making the change, and so forth. Details of past activity can be held within the document, and retain the advantages of passive collection and distribution. What's more, such document-based representations can be presented at various levels of specificity, so that users can access information as it becomes relevant. Such information can also change over time to reflect the progress of activity on the part of individuals; and it can be presented differently to the various collaborators, as is appropriate for their different involvements in the activity and for the way these involvements change over time. The work space, then, holds more than merely the object of group activity, but also becomes a persistent record of that activity.

When we consider presenting past activity information within the shared workspace, the division between synchronous and asynchronous activities becomes less distinct, and we can take this further. A *semi-synchronous* system supports both synchronous and asynchronous work modes. In asynchronous use, the workspace presents past activity information, so as to give an individual awareness of the activities of other participants integrated with the work object itself. In synchronous use, this information is presented as it happens, providing participants with awareness of others' current activities. However, these are not two different modes of the system, but rather are two facets of a single view of awareness information. A semi-synchronous system presents current information on synchronously co-present collaborators, at the same time as representations of past activities by other collaborators who are not synchronously present. The workspace becomes a persistent space in which collaborators can interact, rather like a room in which one can either talk with other people who happen to be there at the time, or leave notes (or other, more passive, representations of one's activity) for those who arrive later.

6 CONCLUSIONS

We have discussed approaches to the critical issue of group activity awareness in collaborative systems. This awareness provides a context for individual activities and thus facilitates group progress. Two existing approaches have been described. The first is an explicit approach which uses directed messaging and the second uses a "strong" notion of roles and activities to convey information to the group about individuals' actions and plans. A third approach is to present shared feedback and results from individual activities to the group at large through the shared workspace. This conveys a continually-updating sense of the actions of individual collaborators and the overall progress of the group.

We have discussed systems which embody each of these approaches. Shared feedback overcomes problems with informational and role-restrictive approaches. In particular:

1. Shared feedback reduces the costs to individuals of information production by collecting information passively and avoiding restrictions on activities.
2. Shared feedback allows participants to look for and extract the awareness information which is most relevant to them.
3. Shared feedback presents awareness information through the shared workspace and linked to it, so that users can (i) find relevant information along with the shared object, and (ii) browse awareness information and the work object concurrently.

Shared feedback can be applied more generally than merely synchronous collaborative systems through the use of persistent, semi-synchronous workspaces and systems.

We are continuing to explore the relationship between explicit and implicit generation of information to support awareness in collaborative systems as part of our research on design principles and generic architectures. It is clear that there is more to learn about each approach and its impact on collaborative work practices.

ACKNOWLEDGMENTS

We would like Judy and Gary Olson for their important role in organising the ShrEdit study, Ian Daniel and Mike Molloy for making it work, and Beth Anderson, Sara Bly, Graham Button, Wendy Mackay, Eliot Miranda and Abi Sellen for comments on earlier drafts of this paper. The development of ShrEdit was supported, in part, by the National Science Foundation under grant IRI-8902930.

REFERENCES

[1] Laurel Austin, Jeffrey Liker and Poppy McLeod, *"Determinants and patterns of control over technology in a computerised meeting room"*, Proc. CSCW'90 Computer-Supported Cooperative Work (Los Angeles, CA, October 1990).

[1] One interface metaphor which frequently crops up in discussions with colleagues is that of the "slime trails" left by snails.

[2] Eevi Beck, *"A Survey of Experiences of Coauthoring"*, to appear in Sharples (ed.), "Computer-Supported Collaborative Writing", Springer-Verlag (in press).

[3] Paul Dourish and Sara Bly, *"Portholes: Supporting Awareness in Distributed Work Groups"*, Proc. CHI'92 Human Factors in Computer Systems (Monterey, Ca., May 1992).

[4] Clarence Ellis, Simon Gibbs and Gail Rein, "*Design and Use of a Group Editor"*, in Cockton (ed.), "Engineering for Human-Computer Interaction", North-Holland, 1990.

[5] Clarence Ellis and Simon Gibbs, "*Concurrency Control in Groupware Systems"*, Proc. SIGMOD'89 Management of Data (Seattle, Wa., 1989).

[6] Robert S. Fish, Robert E. Kraut, Mary D. P. Leland and Michael Cohen, "*Quilt—A Collaborative Tool for Cooperative Writing"*, Proc. COIS'88 Office Information Systems (Palo Alto, Ca, March 1988).

[7] Jolene Galegher and Robert Kraut, "*Computer-mediated Communication for Intellectual Teamwork: A Field Experiment in Group Writing*", Proc. CSCW'90 Computer-Supported Cooperative Work (Los Angeles, October, 1990).

[8] William Gaver, "*Sound Support for Collaboration"*, Proc. ECSCW'91 European Conference on Computer-Supported Cooperative Work (Amsterdam, September 1991).

[9] Jonathan Grudin, "*Why CSCW Applications Fail: Problems in the Design and Evaluation of Organisational Interfaces"*, Proc. CSCW'88 Computer-Supported Cooperative Work (Portland, Or., September 1988).

[10] Christian Heath and Paul Luff, "*Collaborative Activity and Technological Design: Task Coordination in London Underground Control Rooms"*, Proc. ECSCW'91 European Conference on Computer-Supported Cooperative Work (Amsterdam, September 1991).

[11] Mary Leland, Robert Fish and Robert Kraut, *"Collaborative Document Production Using Quilt"*, Proc. CSCW'88 Computer-Supported Cooperative Work (Portland, Or., September 1988).

[12] Lola McGuffin and Gary Olson, "*ShrEdit: A Shared Electronic Workspace"*, CSMIL Technical Report, Cognitive Science and Machine Intelligence Laboratory, University of Michigan, 1992.

[13] Marilyn Mantei, "*Capturing the Capture Lab Concepts: A Case Study in the Design of Computer Supported Meeting Environments"*, Proc. CSCW'88 Computer-Supported Cooperative Work (Portland, Or., September 1988).

[14] Christine M. Neuwirth, David S. Kaufer, Ravinder Chandhok and James H. Morris, "*Issues in the Design of Computer Support for Co-authoring and Commenting"*, Proc. CSCW'90 Computer-Supported Cooperative Work (Los Angeles, Ca., October 1990).

[15] Judy Olson, Gary Olson, Marianne Storrøsten and Mark Carter, "*How a Group-Editor Changes the Character of a Design Meeting as well as its Outcome"*, Proc. CSCW'92 Computer-Supported Cooperative Work (Toronto, Canada, November 1992).

[16] Ilona Posner, *"A Study of Collaborative Writing"*, Unpublished Master's thesis, University of Toronto, 1991.

[17] Deborah Tatar, Gregg Foster and Daniel Bobrow, *"Designing for Conversation: Lessons from Cognoter"*, Intl. Journal of Man-Machine Studies, 34, pp 185–209, 1991.

[18] Walter Tichy, "*RCS—A System for Version Control"*, Software—Practice and Experience, 15(7), pp. 637–654, July 1985.