Distortion Oriented Workspace Awareness in Collaborative Writing

Philip Weir

November 1, 1996

Abstract

In order to use groupware applications effectively a sense of workspace awareness is required. Distortion oriented visualisation techniques, such as fisheye lenses, have been proposed as a means of achieving this awareness. These techniques have been demonstrated in point systems but there is a need to test their effectiveness in 'real' applications. DOME is a collaborative writing system written to evaluate the effectiveness of these techniques in a larger context. Some preliminary usability studies have been conducted to eliminate problems and give an indication of the effectiveness of the technique.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Workspace awareness – the information about who you are working with, where they are working, and what they are doing – provides important information for people working together in groupware systems. Even in our normal non-computer work this information proves useful, letting us know what the rest of the group is doing, helping us avoid duplication, and enabling group members to assist each other. In such a setting workspace awareness information is obtained by glancing around but when we use a computer system, this information can be more difficult to obtain. Computer screens provide us with a small window onto the workspace, and our peripheral vision is severely restricted.

Distortion-oriented visualisation techniques address the problem of restricted screen real-estate by altering the visual representation of the workspace, so that local information is available along with peripheral information. Point systems have demonstrated the feasibility of these methods for providing workspace awareness, however demonstrating feasibility does not guarantee the usability or effectiveness of the technique. A point system merely demonstrates a technique, presenting it in isolation, and often in a contrived manner that does not reflect how it would be incorporated in a realistic application. This project aims to address this problem by incorporating a distortion oriented visualisation technique–a fisheye lens–in a 'real' system. A further goal of this project is to evaluate the fisheye lens technique, to determine its effectiveness in providing workspace awareness, in the context of a 'real' system.

Comparative studies in CSCW are not common, with most systems being evaluated based on the user's reactions to it – a try it and see approach. This is often necessary because the systems being evaluated are novel and there may be nothing to compare them with. Comparative studies offer the advantage that a base line can be compared against, and such a study is needed for workspace awareness techniques. Allowing experimenters to evaluate various combinations of workspace awareness techniques, in a variety of situations, may indicate which combinations are best. A further goal of this project is to equip DOME[1] with mechanisms that allow evaluators to control experiments with the system.

1.1 Report Structure

Chapter 2 presents a discussion of the background areas of research that this project draws on. The fields of Computer Supported Cooperative Work (CSCW), distortion oriented visualisation, collaborative writing, and Human Computer Interaction (HCI) are all explored. Chapter 3 uses this backgound information to present a design for an ideal collaborative writing system, incorporating distortion oriented visualisation techniques to provide workspace awareness, and support experimentation. Chapter 4 presents DOME, a distortion-oriented collaborative writing system based on these design criteria. Design decisions and implementation problems are discussed. Chapter 5 reports on a preliminary evaluation study conducted to highlight any problems in the interface,

[1] *D*istortion *O*riented *M*ulti-person *E*ditor

and to observe DOME in use. Chapter 6 closes the report with a look at future work, and a summary of the project's achievements.

Chapter 2

Background

This chapter explores the related research areas that this project draws on. Figure 2.1 indicates how the relationships between these areas of research can be viewed.

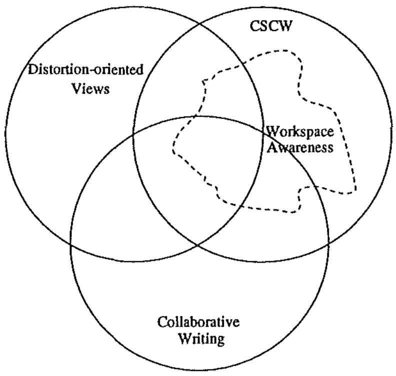


Figure 2.1: The relationship between distortion oriented views, CSCW, and collaborative writing.

This chapter is structured as follows: Section 2.1 reviews research on Computer Supported Cooperative Work (CSCW). Section 2.2 examines distortion oriented visualisation techniques, and exemplar systems. Collaborative writing is scrutinised in section 2.3, where system requirements are considered and example systems are reviewed. Section 2.4 discusses issues of design related to the user interface, groupware, and the design of tools.

2.1 A Review of CSCW

This section reviews the multidisciplinary field of CSCW and begins by presenting an overview of what CSCW is. CSCW is a broad field, so section 2.1.2 presents some taxonomies that classify CSCW in different ways. Section 2.1.3 explores another taxonomy based on the application domains in the field. Two important issues in CSCW – WYSIWIS and Workspace awareness – are then examined in sections 2.1.4 and 2.1.5.

2.1.1 Introduction to CSCW

CSCW is concerned with how groups interact, and in particular, how they work together when they use technology (Ellis *et al.*, 1991; McCarthy & Monk, 1994). CSCW is both a theoretical and applied discipline. Theories come from areas such as sociology, psychology, linguistics, and anthropology, and applications from computer science. Systems written to support groups, called *groupware*, include messaging systems, collaborative writing systems, meeting rooms, conferencing, and coordination systems. Ellis *et al.* (1991) define groupware as follows:

> computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment.

Until recently most groupware systems have been research projects, but increases in computing power, network availability, and user demands, should see an increased presence of these systems. Traditional multiuser systems such as database management systems have been available for some time, and can be seen as a form of groupware. They tend to present an impression that there is a single user on the system. Groupware, in contrast, aims to alert all users to the presence of others users.

Humans are social beings, and work is a social undertaking (Kraut *et al.*, 1988). Artifacts of work may be shared or discussed by co-workers, and it is common for work to be undertaken in groups (Kraut *et al.*, 1988). For instance, an author may consider she writes "alone", others others such as editors and proofreaders are often involved in the process. If a user wishes to use a groupware system outside of a group, they should not be prevented from doing so (Cockburn & Jones, 1995).

The *Work* aspect of CSC*W* is used in its most general sense, and includes activities such as learning (CSCL), and entertainment (multi-player games). Rather than being restricted to just computer-based systems, systems can use technology such as video (Francik *et al.*, 1991; Bly *et al.*, 1993; Brittan, 1992). Most researchers use the term 'CSCW' in its most general sense, and it has come to mean more than the sum of its parts – Synergy is a property of effective collaboration.

Single-user software systems and their user interfaces are difficult to develop, and Grudin (1990) warns that groupware systems are much more difficult. Groupware systems encounter extra political and social issues, and our intuitions for building groupware systems are poor. Hollan & Stornetta (1992) stress the importance of looking beyond emulating human-human interaction. The CRUISER system's goal of providing face-to-face communication was underutilised as participants preferred the real face-to-face when it was available (Fish *et al.*, 1993).

2.1.2 Classifications of Groupware

Various taxonomies of CSCW have been presented in the literature (Ellis *et al.*, 1991; Dix *et al.*, 1993), and the following sections will present a selection of these.

Time/space matrix

A common starting point in CSCW is to classify systems into quadrants of a time/space matrix like that shown in figure 2.2. This framework is in frequent use, and provides a good initial framework,

	Synchronous	Asynchronous
Co-located	Face-to-face conferencing Electronic meeting rooms	Coordination and Project Management Systems
Distributed	Video Walls Desktop conferencing	Enhanced email Letters

Figure 2.2: Time-location perspective of CSCW

but is relatively simple. Dix *et al.* (1993) present an enhanced model, shown in figure 2.3, which considers access to the shared data, and indicates that location is really only an issue in concurrent synchronised collaborations.

	co-located	remote
(a) concurrent synchronised	meeting rooms	video conferences, video-wall, etc
	Shared work surfaces and editors shared PCs and windows	
(a/b) mixed	co-authoring systems, shared calendars	
(b) serial	argumentation tools	
(c) unsynchronised	email and structured messages electronic conferences	

Figure 2.3: Refined time/space matrix

Entity Relationship Framework

Figure 2.4 shows the entity relationship framework for cooperative work (Dix *et al.*, 1993). This

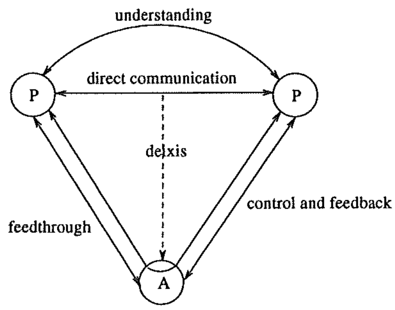


Figure 2.4: The Entity Relationship Framework.

model represents users (labelled P) who can communicate directly and build up an understanding. They can control an artifact (labelled A), and receive feedback from it. Actions made to the artifact are also fed through to other participants. Deictic reference is indicated as reference, in the direct communication between participants, to the artifact. Breakdowns in these deictic references have been shown to be an important factor in groupware systems, as was demonstrated in Cognoter (Tatar *et al.*, 1991). Participants are indirectly interacting through the shared artifact, and it is important that constant feedback and feedthrough is provided, and that this is provided in real time..

Tasks and Environment

Ellis *et al.* (1991) also present a task and environment spectra. Figure 2.5 shows spectra representing the commonality of tasks, ranging from unrelated tasks running on the same machine, to highly related software engineering tasks. There is also a shared environment spectrum indicating the amount of sharing of the data space.

Control

A fourth framework relates to the control of the tasks. Tasks may prescribe to a particular way of working, as in workflow systems, or support unstructured activities as in brainstorming tools. Control may be explicit, as a floor control or turn-taking policy, or be implicit and controlled by the users using a social protocol.

There have been many attempts to model work. These have often involved theories such as speech acts (Flores *et al.*, 1988). It is important to realise that a model involves simplification, and

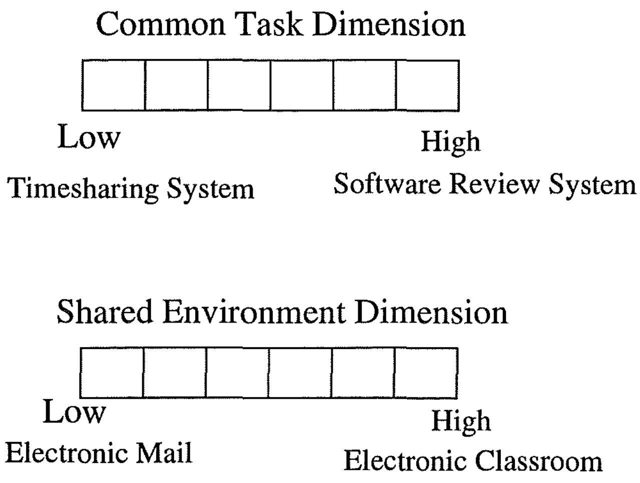


Figure 2.5: Task and Environment spectra for Groupware.

to ensure that important features are not being omitted. The model should reflect the way people work, and not force users to work in ways that are unnatural (Flores *et al.*, 1988). The model should also cope well with any exceptions (Grudin, 1990). The model used in the Coordinator (Flores *et al.*, 1988) for instance compromised the ability for users to informally negotiate, resulting in it being unusable by most organisations.

Discussion on Frameworks

The frameworks presented provide a method of talking about groupware systems, but other social factors that affect all these interactions need to be considered. Grudin (1990) warns that in designing groupware social acceptability must be considered. Often systems are designed for management's benefit, with less consideration given to the users who will be required to provide extra effort to make such systems work. Systems should not 'straight-jacket' users, but allow for non-standard, or unexpected events, and allow for users to be creative in their use. He also notes that evaluation of groupware is a difficult task, and our intuition is not a good guide for building systems.

2.1.3 Groupware Application Domains

This section will present some examples related to particular application domains, based on the classification of Ellis *et al.* (1991).

Messaging systems

Messaging systems are often based on e-mail and bulletin-boards. Extensions to these tools focus on providing filtering and structuring The Information Lens (Malone *et al.*, 1988) was an attempt to apply structure to electronic communications by using templates to filter and categorize the information.

E-mail is the most prevalent groupware application currently. In categorising or filtering it is important that not too much reliance be placed on information provided by the sender (Grudin, 1990). There may be errors in categorisation. or because of the disparity of costs and benefits, the sender may have omitted to structure the message as there is no benefit for them.

In interactions it may be possible to derive some sort of 'structure'. There have been attempts to achieve this in filtering of e-mail, in the Information Lens (Malone *et al.*, 1988)), and in GDSS systems where 'threads' of conversation are structured to make it less cognitively demanding on the user to locate information, for instance in the gIBIS system (Yakemovic & Conklin, 1990).

Multi-user editors

Multi-user editors allow users to work together in editing documents. A variety of such systems have been built, some providing synchronous, some asynchronous, support. Section 2.3 investigates such systems in depth.

Group Decision Support Systems / Meeting Rooms

Meeting room and Group Decision Support systems are designed to explore unstructured, or semi-structured problems such as those discussed in meetings (Tatar *et al.*, 1991; Stefik *et al.*, 1987). They often provide support for activities such as brainstorming, tools to analyse or structure group discussions, and whiteboards. The aim of these systems is to improve the decisions, by improving the speed, and/or the quality of the results. Some systems involve specially constructed meeting rooms, with computers recessed so that face-to-face contact can still occur (Mantei, 1989). Groupsize is also an important consideration in these system, especially in their ability to cater for large groups.

Computer Conferencing

Computer conferencing systems use the computer as a communications medium (Ellis *et al.*, 1991) and can provide the facility to share artifacts in real time by means of a shared workspace.

Realtime conferencing involves the synchronous use of a shared workspace. Systems may be collaboration "transparent" and provide a "wrapper" for a single user application, and provide for turn-taking. They may be collaboration "aware", with the application being specially designed to cater for groups. The first approach is technically easier, and allows for the use of existing single-user applications, while the second approach can provide for much richer interaction.

Computer teleconferencing typically involves video technology and conference calls, and often requires special rooms. The Videowall (Bly *et al.*, 1993) provided support for social awareness through a media space, and the CRUISER (Fish *et al.*, 1993) aimed to support informal interactions.

Desktop conferencing is a combination of realtime and computer teleconferencing, providing video access through a computer.

A lot of video-based systems attempt to mimic face-to-face interactions. There has been criticism that this makes these systems inferior to '*real*' face-to-face interaction (Hollan & Stornetta, 1992). Humans are able to read extremely subtle cues from each others faces, but these do not transfer well to video. Work on providing '*eyegaze awareness*' (Ishii & Kobayashi, 1992), for instance can provide some of these cues. This technique is currently limited to a pair of users, but in the future perhaps instead of providing a full video image, important features can be derived. Facial feature analysis is an active research area, and this may enable the 'semantics' of the gestures to be analysed and presented.

Real-time interaction is becoming more readily available with the recent proliferation of powerful personal computers and the availability of increased bandwidth on networks. Most realtime systems are currently only practical on LANs, unless special fast links are installed. With improvements in network technology, such as ATM, the use of WAN's is expected to become more practical.

Intelligent Agents

Intelligent agents have been built to act as facilitators in meetings, help in organising and consolidating ideas from brainstorming, and for filtering email (Chen *et al.*, 1996).

Coordination Systems

Coordination systems aim to coordinate activities in an organisation. Models based on *forms* that are passed around, *processes* used, *conversation*, and *communication-structure* have been used to model organisations.

Organisations "seem" to follow general patterns, and coordination systems use these patterns to model the organisation and to simplify routine tasks. Exceptions to these patterns often can not be codified, and it is important that the system does not interfere or slow them down because they are outside the model. It may be important for such models to adapt to changes in the organisational structure, or the changing needs of users and the organisation.

2.1.4 WYSIWIS

WYSIWIS (What You See Is What I See) provides identical views to all participants. Although this reduces problems associated with deixis, the requirement that all views be identical constrains users who may wish to work on seperate tasks.

Dix's Entity-Relationship framework, shown in figure 2.4, shows that in communicating with each other, participants make reference to artifacts of work. Deictic breakdowns occur if the representations, or state, of these artifacts are different for each participant. The Colab system (Stefik *et al.*, 1987; Tatar *et al.*, 1991) demonstrated the importance of keeping actions and artifacts synchronised in real-time interactions. Users require constant feedback in an interactive fashion, whereas in Colab transmissions were delayed until users had finished their actions. Deictic references broke down, as user's screen were different. The continual update and transmission is expensive in regard to network usage, but is necessary to avoid collaborative breakdowns. As networks work at higher speeds and optimisations in the form of compression technology become widespread this may become less of an issue.

2.1.5 Workspace Awareness

Dix *et al.* (1993)'s entity-relationship framework in figure 2.4 shows an interaction occuring between participants *through* the artifact. The artifact, or collection of artifacts, can be considered as a workspace. Consider a group of four working around a table in a brainstorming session, using sheets of paper to write or draw on. These sheets of paper are considered artifacts in the workspace.

A range of types of awareness have been identified as being useful in groupware systems, and in collaborative work in general. These include the following:

- Informal Awareness: Information about who is around, and what they are doing.
- Social Awareness: Whether someone is paying attention, their emotional state, where they are looking, facial expressions.
- Group Structural Awareness: Roles and Responsibilities of members of the group.

Examples of systems that support these types of awareness are available in the literature (Bly *et al.*, 1993; Ellis *et al.*, 1991; Dix *et al.*, 1993; Ishii & Kobayashi, 1992).

When we consider workspace awareness we are interested in factors such as the following.:

- Who is in the workspace.
- Where they are.
- What they are doing.

In the brainstorming example, this information is available to group members by glancing around, and through peripheral vision. In a computer system, our view is restricted to a smallish rectangular window. Some of this information may be available through other channels, such as audio if in close proximity, or if an audio link is available.

Workspace awareness also contains other dimensions, including a temporal aspect – when something happened. For example if we leave the workspace, when we return, we may wish to catch up on what has happened.

Various methods for providing workspace have been discussed in the literature, and include techniques such as gestalt views, telepointers, teleselections, and multi-user scrollbars, Fisheye lenses have also been demonstrated in point systems as another available technique (Gutwin *et al.*, 1995; Greenberg *et al.*, 1995, 1996).

2.2 Distortion Oriented Views

Distortion oriented visualisation is an active area of research in the fields of HCI and Graphics. These distortion oriented techniques address the problem of visualising large amounts of data, and attempt to maximise the effectiveness of the available viewing space. For instance, in viewing a large data space on a computer screen, such as a map of the United States, we have to compromise between viewing the entire map, and a region in detail. In viewing large amounts of data, the usual solutions are as follows:

- shrinking the data so it will fit;
- sliding a window over the data;
- zooming in to, or out of, the workspace as required.

Shrinking the data provides a global view of the data, but without much detail, whereas sliding a window provides local detail, but without global context. Zooming allows switching between providing local content and global context, buts only gives one at a time. We may have a global and local view in separate windows, such as in a gestalt viewer, however there is a need to integrate the two views, and being in seperate windows consumes valuable screen space. If we want both, distortion-oriented visualisation techniques provide a variety of solutions.

A global view of the data could be provided in a window and a *magnifying lens* provided so that an area could be seen in some detail, as is shown in figure 2.6. The magnified area is shown in detail, but context information is lost because it is occluded by the lens. Those points of data just

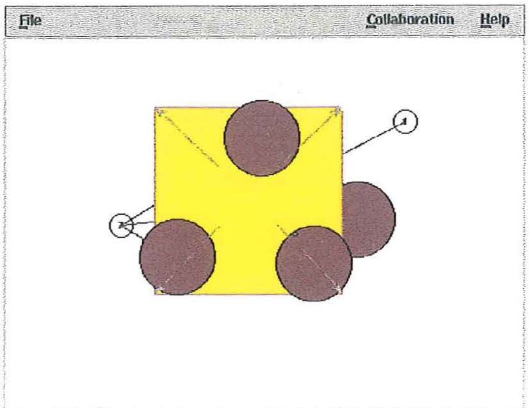


Figure 2.6: A Magnifying Lens View

outside, and neighbouring, the magnified area are hidden behind the magnified view. A solution is to provide an *offset lens* which is a similar technique with the additional facility to offset the lens and magnified regions. An example of this is shown in figure 2.7.

The perspective wall (MacKinlay *et al.*, 1991) is a technique which slides a window around the workspace but also shows context by presenting neighbouring data as a wall receding into the

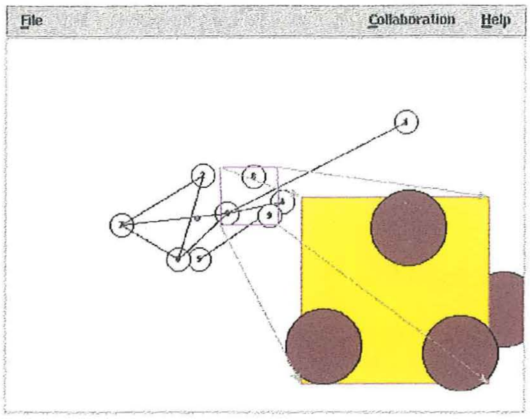


Figure 2.7: An Offset Lens

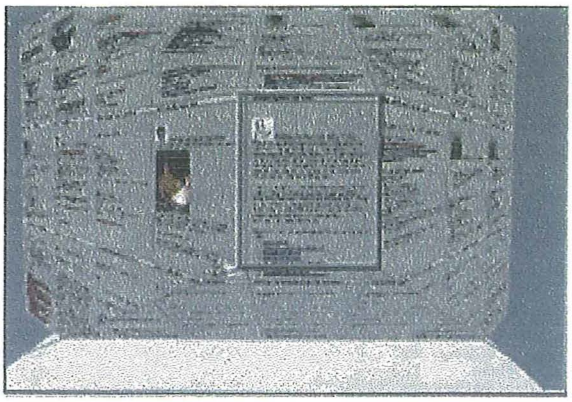


Figure 2.8: A Document Lens

distance. Variations on this, such as the Document Lens, provide a perspective in two dimensions. An example of a Document Lens is shown in figure 2.8.

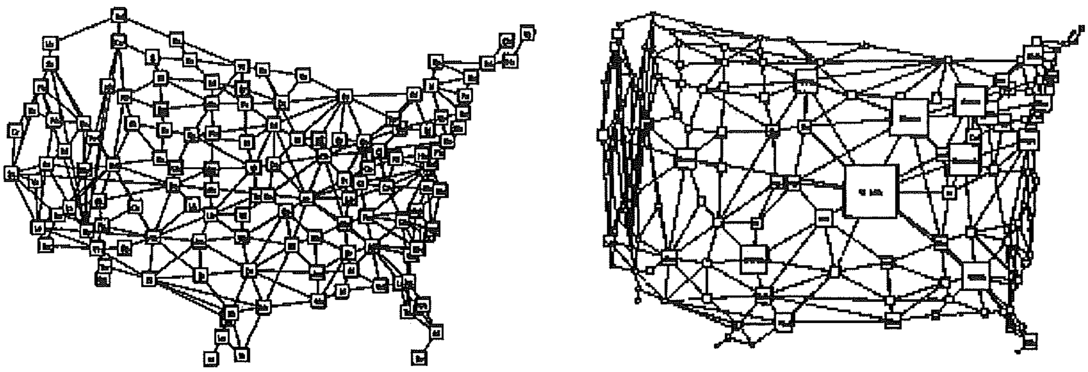
Fisheye views (Furnas, 1986) consider two factors in applying a distortion to a workspace:

- the distance from the focal point
- an *a priori* level of importance

Figure 2.9 shows a map of the USA, with an undistorted view on the left and a view with a fisheye lens applied on the right. The figure on the right has a focal point at New Orleans, causing points closer to the focal point to appear as larger, and those further away as smaller. If other information, such as population density, was included in the undistorted view, then this would be considered in applying the fisheye distortion.

A useful taxonomy of these techniques appears in Leung & Apperley (1994), along with a unifying theory. This theory uses an analogy to a frame mounted rubber sheet, where stretching or compressing points of the sheet produce the desired distortion-oriented presentation effect.

The fisheye and perspective wall techniques present global context and local content in a single view, so the user does not need to mentally integrate views to build an accurate picture. Applications that have incorporated fisheye views include source code viewers (Furnas, 1986), map viewers (Sarkar & Brown, 1994), database structure diagrammers (Churcher, 1995), network



(a) undistorted view of USA

(b) Map of USA with fisheye distortion applied

Figure 2.9: A map showing a fisheye lens applied

navigation (Schaffer *et al.*, 1996), and in groupware to provide workspace awareness (Tatar *et al.*, 1991; Greenberg *et al.*, 1996).

2.2.1 Collaborative Distortion Oriented Views

Distortion oriented techniques are now beginning to be seen in groupware systems, mainly as proof of concept systems.

Fisheye lenses have been applied in groupware systems to indicate social activity (Ackerman & Starr, 1996), and in point systems to provide workspace awareness (Greenberg *et al.*, 1995, 1996).

The Groupkit distribution (Roseman & Greenberg, 1992) includes a number of point systems that deal with issues of workspace awareness. A *fisheye text viewer*, shown in figure 2.10 is included in this distribution, and demonstrates the application of a fisheye lens in providing workspace awareness in group situations. Each user's focal point is represented by a fisheye lens, enabling users to observe where they are in the document (global context), and what they are doing there (local detail).

DOME, the system described in this report, extends and iterates on these preliminary works.

2.3 Computer Supported Collaborative Writing

Computer Supported Collaborative Writing is an active area of research in CSCW and forms the basis of many CSCW systems. Engelbart's NLS system was an early collaborative writing system, that also first demonstrated the use of teleselections and telepointers in the early 1970's.

In reviewing collaborative writing, how groups of people interact in producing documents is of interest. Research into the way people write together (Posner & Baecker, 1993) has produced a taxonomy based on the following:

- *roles*, such as writer, reviewer, editor;
- *activities*, including brainstorming, writing, final edit;
- *document control methods*, including shared or central control, relay control, or independent control;
- *writing strategies* such as single writer at a time, a scribe, joint writing, or seperate writing.

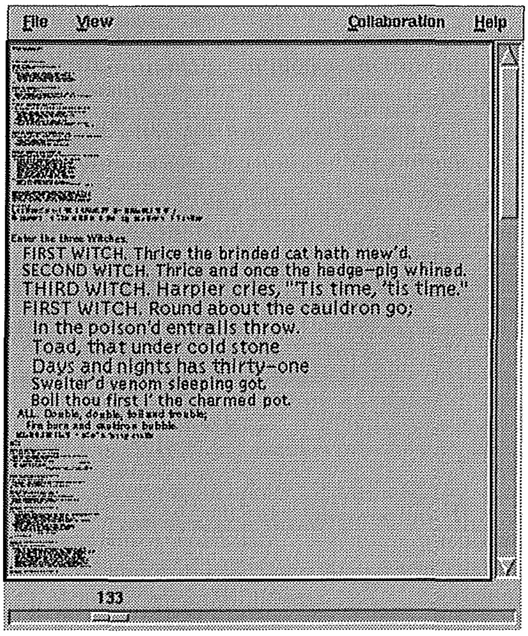


Figure 2.10: The fisheye text viewer

The type of document being written may also influence factors such as the document control and writing strategies. For instance a hypermedia document lends itself to division along the lines of hypermedia nodes, and certain tasks like software engineering may require a centralised control.

This taxonomy provides a useful structure that can encourage discussion about collaborative writing, but does not fully cover the dynamic nature of writing. Each group tends to have their own ways of working together, and participants often move between various roles and activities. Systems have been written that enforce explicit roles, and while this may work well for some groups, many find the experience restricting. Jones (1992) suggests providing support for the following major writing activities:

- *Planning*: where ideas are generated and organised, and goals are set.
- *Translating*: where the text is generated.
- *Review*: where the text is improved.

Certain authoring tasks, such as those in software engineering, may benefit from a prescriptive model rather than relying on social protocols. However, most authoring is of a less formal nature and people need to fluidly change their roles and activities, and the system needs to support this requirement rather than enforce an explicit model.

Many examples of collaborative writing are available in the literature, including AUGMENT (Engelbart, 1988), Quilt (Leland *et al.*, 1988), GROVE (Ellis *et al.*, 1991), MILO (Jones, 1992), SASSE (Posner & Baecker, 1993), ShrEdit, WinFold (King & Leung, 1994), SEPIA (Thüring *et al.*, 1995). Many of these provide support for certain aspects of the writing process, with most providing either synchronous or asynchronous support. Dix *et al.* (1993) identify a need for bridging gaps between synchronous and unsynchronous use, and synchronisation of the data store.

2.4 Issues of Design

"The daily experience of using computers far too often is still fraught with difficulty, pain, and barriers for most people". (Kapor, 1991)

There are many examples of design failures in everyday life, such as the controls of a video recorder remote control (Norman, 1988). Graphical User Interfaces (GUI's), in themselves, do not make a good design, and there are many examples of bad design in computer systems.

Designers need to "focus on people and their situations: how people experience software; what they do with it; and the larger situation in which they encounter it" (Winograd, 1995). Historically most of the effort in designing computer systems has focused on creating efficient implementions for the computer. Cheaper, more powerful personal computing power, competition, and general purpose applications have helped shift the focus to the user.

Section 2.4.1 discusses issues in the interactions between humans and computers. Section 2.4.2 examines the issues that arise when groups are working together with computers. Not all applications will necessarily meet the needs of all users. The provision of general purpose tools that can be used in a variety of situations can help in this, and this is discussed in section 2.4.3.

2.4.1 HCI Issues of Design

HCI (Human Computer Interaction) is concerned with the interface between the human user and the computer system. Many guidelines have been developed for HCI (Norman, 1988; Thimbleby, 1990; Dix *et al.*, 1993; Nielsen, 1993), and this section will present a brief overview of the features they have in common. Most guidelines suggest similar principles, and only differ in their level of detail, and the terms they use.

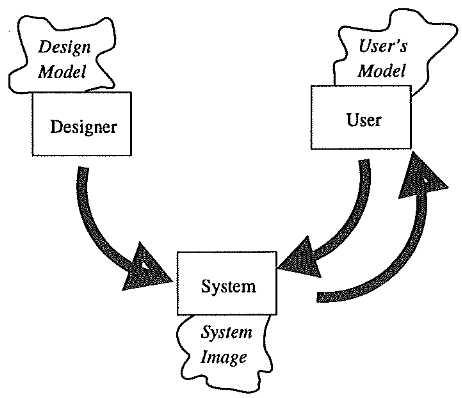


Figure 2.11: A Conceptual Model of Design

Norman (1988) discusses general principles of design and developed the conceptual model shown in figure 2.11. The arrows in the figure indicate mappings between the participants and the system; it is at these mappings that breakdowns often occur. The perspectives of the human participants are important in the model. The designer has an image of the model that they wish to present to the user; a design model. This is presented to the user through the system, as the system image. The user uses the model formed to 'drive' the system, and obtains feedback via the system image. Problems in any of these mappings can result in an inappropriate model of the system being formed by the user, or an inappropriate system image being presented. The aim is to make these models as close as possible to each other. Thimbleby (1990) writes 'The designer is obligated to ensure the user constructs and can use an appropriate model of the system'.

Norman (1988) presents very general guidelines, and give a useful context for design. Dix *et al.* (1993) present their usability principles in more detail, using the three categories of learnability, flexibility, and robustness. The principles that support the categories are as follows:

Learnability how easily new users can begin using the system, and achieve maximal performance. The principles that support learnability are as follows:

- Predicatability: whether the user can tell the effect of their next action;

- Synthesizability: if the user can determine the effect of past actions;
- Familiarity: whether other applications or previous knowledge can assist the user;
- Generalizability: if knowledge can be extended;
- Consistency: similarity in input/output behaviour.

Flexibility the number of ways in which the user and system can exchange information. The principles that support flexibility are as follows:

- Dialogue initiative: allow the user choice of action;
- Multi-threading: more than one task operating at a time;
- Task migratability: the ability to choose whether the user or system controls a task;
- Substitutivity: allowing equivalent values to be substituted, such as using inches instead of centimetres;
- Customisability: whether the interface can be modified.

Robustness supports the user in determining whether their actions and goals succeeded. The principles that support robustness are as follows:

- Observability: allows the user to determine the internal state of the system from the external representation;
- Recoverability: allows the user to correct errors;
- Responsiveness: how fast the user perceives the system to be;
- Task conformance: whether all tasks required are available, and in an understandable form.

This is but one of many sets of guidelines to keep in mind when designing a user interface. In general the user should feel comfortable in the environment, have a degree of control, receive good feedback, and understand what is going on. The system should be predictable and forgiving, and the user should be able to determine how to achieve their goals.

2.4.2 CSCW Issues of Design

In designing for groups extra requirements such as feedthrough, as shown in figure 2.4, must be included for synchronous use. In single-user systems users are able to form a model of the system by observing the system's responses to their actions. In groupware systems the actions of others will affect the workspace, so it will be more difficult to determine a causal relationship. Workspace awareness can be of use, in assisting users to form a causal model, because it provides extra information about the activities of coworkers. The entity relationship framework in figure 2.4 captures the communication paths of interest in synchronous activity. The usability principles from section 2.4.1 that are affected include the following:

- Predictability – Actions in the single-user case are from the user, or system generated. With multiple users, those actions we did not generate are either system generated or from other users. There is a need to distinguish between these two sources for uses to be able to form an accurate model.

- Responsiveness – In synchronous groupware each action may have to be broadcast to all participants in the conference. It is no longer just communication from user $\leftrightarrow$ system, but extends across all users. This increases the complexibility, and also the duration of events. Speed is an issue of concern in groupware systems, as breakdowns can occur

- Familiarity – It is uncommon to have others present in one's application., and it may take time to adapt to this concept.

- Recoverability – changes can be generated now from multiple sources, and recoverability becomes much more complex. In a single-user model the state of the system can be modelled as a state-transition-diagram, and recovery involves returning to a previous state. In group work a users last action may not be the last 'global' action, and the actions of others may make recovery impossible.

Groupware is in need of good design principles, and Cockburn & Jones (1995) presents four principles to guide design:

Maximise personal acceptance : It is often difficult to get the entire group to adopt a tool at the same time, therefore tools should be usable by single-users, with additional benefits when used by the group. To be adopted, tools must be able to compete with similar single-user tools, and provide features such as a solid interface, and robustness.

Minimise the requirements imposed on users : Systems such as the Coordinator (Flores *et al.*, 1988) required that senders provide extra information for the benefit of the receivers. Such information may be obtainable from other sources at less cost to the sender.

Minimise constraints imposed on users : And reflect how users would like top work. Theories of work are often incomplete and can result in the inclusion of implicit models (Tatar *et al.*, 1991).

Maximise external system integration : General purpose tools and frameworks are good software engineering practice, and make further developments easier.

2.4.3 The Design of Tools

In groupware systems tools have been provided in the following ways:

- The tools can be embedded into the application. Extra work is required to embed the tools, and if a new tool is required, work is required to integrate it. Standard tool interfaces are available in some application domains.

- Tools can be used in 'stampsheet' or 'rooms' environments (Stefik *et al.*, 1987). In this approach all tools are written to operate in the environment. The writing application, and any supporting tools, are written for the environment. New tools are written for the environment, rather than being integrated into the application. Rooms concept may also provide a structure external to the tools allowing structural links across tools.

Computer-based tools should be accessible, and offer a natural model of use to the user. Users move easily between various phases and activities in their everyday work, and computer-based systems need to provide this ease of use if they are to compete. If the effort of acquiring, or using, a tool is too great, users will seek alternatives (Neuwirth *et al.*, 1990). The ease with which users can manipulate natural objects in the real world, and transfer seamlessly between tasks, is often not realised in computer systems. Seams are often generated in computer-based systems because tools are provided in separate windows, or transition to these tools requires extra effort (Tatar *et al.*, 1991). Good design should aim to minimise these problems of accessability and seams. Brown (1996) suggests that:

> "The best design lets us reach through computers into the world, allowing us to focus on creating value, not manipulating tools."

Tools are for purposeful endeavours, to achieve some goal, or to solve a problem. A single tool or method may be insufficient to solve a problem, and strategies such as decomposing the problem into subtasks must be used (Anderson, 1990). The ability to use general-purpose tools, and link parts of the problem or solution spaces to each other, offers users the flexibility to structure their work according to their needs, rather than following a prescribed model. Engelbart (1995) suggests "every knowledge object, down to characters, should be uniquely addressable, and referenceable anywhere in the hyperdocument system." Providing a palette of tools, and the ability to combine them in novel ways, allows a natural way of working, that will not get in the way, as more rigid approaches may.

Application domains, such as software engineering, have specialised tool environments and interfaces. Interfaces such as Portable Common Tool Environment (PCTE) allow tools to be written to conform to a common interface (Sommerville, 1992), and operate together. Object-orientation techniques, such as OLE and CORBA, can allow tools to be shared by providing a common interface. A uniform method for the supply of tools can ease the problems of tool integration, by allowing users the freedom to choose the tools they require.

The problem with providing general purpose tools lies in providing seamless integration of, and access to, the tools.

2.5 Summary

This chapter has presented a background of the areas that are involved in this research project. The review of CSCW presented various taxonomies to describe the field, and examples of groupware applications. Collaborative writing plays a central role in CSCW; with much group activity involving writing, and it encounters most of the major issues in this field. Users working together on a shared workspace were shown to need workspace awareness for a variety of reasons. The requirement that workspace awareness provide users with detailed information about coworkers' activities requires a more effective use of the available screen space. Distortion oriented visualisation techniques present a solution to this problem, and some examples of demonstrational groupware systems using these techniques have been discussed.

The next chapter looks at design issues and draws on this background information to present a design for a collaborative writing system using distortion oriented visualisation techniques to provide workspace awareness.

Chapter 3

Towards Ideal Collaborative Writing Tools

Background information is important because it helps us understand the issues that have been encountered previously, and to avoid repeating past work, or mistakes. The discussion of collaborative writing in chapter 2 indicated that the writing process involves more than just the generation of text. A writing tool must support all the phases of writing, including, but not restricted to, the construction of text documents. This chapter will present the design of an *ideal* writing system, using the background knowledge presented in chapter 2.

Section 3.1 investigates the requirements of single-user computer-based writing systems. Section 3.2 examines *collaborative* writing requirements, so that the group can be supported, and focuses on the provision of workspace awareness. An aim of this project is to evaluate workspace awareness techniques, and section 3.3 discusses how support can be provided for experimenters.

3.1 Writing Requirements

This section investigates the functions, utilities, and activities that are useful in a single-user writing system. The phases of writing, presented in section 2.3, provide a useful context for discussing the support required in the various activities of writing. Section 3.1.1 investigates the support of these phases. The basic functionality required for the editing of text is examined in section 3.1.2, as this is still a central facet of any writing system. Groups determine how they operate, who does what, and when things are done. Aspects relating to how the document is controlled, and temporal aspects of writing, are explored in section 3.1.3. Writers need to be able to comment about the document, as well as writing it, and section 3.1.4 explores annotation mechanisms. Section 3.1.5 examines the need for navigational assistance in large or complex documents.

3.1.1 Support for the Phases of Writing

The phases identified in section 2.3 presented a "loose description of the writing process, that is recursive and not necessarily sequential" (Jones, 1992). Each of these phases indicates certain activities and related functionality, however in providing tools to support these activities it is important that they can function in the fluid manner in which writers operate.

The type of document being created will strongly influence the tools that will be appropriate. Software engineering has suites of tools and techniques available, such as CASE tools, while for writing reports tools such as spelling checkers, grammar checkers, and structuring tools are more appropriate.

The following sections examine the requirements for the support of the phases of writing in detail.

Planning

Planning consists of three subphases: generating, organising, and goal setting (Jones, 1992). In generating, ideas may come from memory, or be the result of research, or brainstorming. Tools for browsing databases, library systems, and the internet can assist writers in locating information. Brainstorming, and idea organisation tools, assist in getting ideas written, and discussed. Idea organisation tools, such as gIBIS (Yakemovic & Conklin, 1990), can help in the decision-making process, the documentation of those decisions, and the systematic investigation of alternatives. The structure of a document is important in report writing, and tools to help develop, and manipulate the structure of the document, are useful. Outline editors, such as GROVE (Ellis *et al.*, 1991), allow detail to be suppressed so that writers can concentrate on manipulation at the structural level. Presenting appropriate high-level representations enables global changes to be made in ways that would be more difficult with all the 'low-level' text present, and working with an appropriate representation lets writers focus on the task (Jones, 1992).

Tools for planning include both bottom-up and top-down approaches. Writers may jot down snippets of information, in a brainstorming manner, and later link this text together to produce the final document.

Translating

In the translation phase the plan is used as a guideline to generate text. Access to information from this phase should be available, at least as reference. Many of the features available in current word-processing systems will be useful, and these are discussed later in a look at the basic editing requirements in section 3.1.2.

Review

The review phase is concerned with improving the quality of what has been written. Improvements may be made to the plan, the document, or any aspect of the process.

Writers can draw and write on a printout produced from a wordprocessing system, in order to provde comment and markup information.

When a wordprocessor is used, to review the work a hardcopy may be produced, and comments written on this. Items and concepts can be linked easily by drawing lines between them. Items can be 'crossed-out', but they remain visible, allowing their contents to be easily recovered. This activity indicates that the following may be useful:

- Checkpoints – the printed copy is a version of the document *frozen* at some time.
- Comments – commenting about the written text on a hardcopy does not affect the main document. The ability to write *about* rather than *in* the document is very useful.

Computers can offer other techniques, such as hypertext, and support for versions, which are not available in manual systems. The requirement for users to be able to fluidly switch between tools in a seamless manner was noted in section 2.4.3. If barriers are present, the likelihood of the tools being used is reduced.

3.1.2 Basic Editing

The features available in current word-processing systems must also be present in a collaborative writing system. Section 2.4.2 discussed the need for CSCW to take a reflexive approach in providing tools; if the basic functionality provided in single user tools is not present in the groupware tool, users will revert to the single user tool. Groupware features are 'extras', and such tools may also be used by writers working alone. The minimal features that should be provided include the following:

- file access and storage;

- selection of text;
- cut, copy, paste, delete operations;
- undo;
- search and replace;
- scrolling.

This is a minimal set, and particular groups may need extra features as an enticement.

Many users have learnt a particular editing system, and will have expended much cognitive effort in learning the native keybindings. Although it is faster to learn a second editor than learning from scratch (Anderson, 1990), providing the familiar keybindings is a much more user-friendly approach. Direct mappings between the new and familiar systems may be impossible or difficult. In emulating another system, all functionality of the old system may not be available, or new functionality may not be utilised.

Selection

Many word-processing systems use direct-manipulation (Williams, 1983; Shneiderman, 1987), and conduct their operations using a selection–action, pattern. Items such as characters, words, sentences, paragraphs may be selected, and an operation applied to that item. Operations that can be applied to the selection should include at least – cut, copy, paste, and delete.

Recoverability

Recoverability enables users to explore the interface trying things out, with the knowledge that they can restore the system if something goes wrong (Norman, 1988; Thimbleby, 1990). This can be achieved by repeatedly saving the document to disk, but this requires an explicit action by the user. The most common approach is to provide an *undo* mechanism which returns the system to a previous state. Two varieties of undo are popular.

- history – this maintains a list of previous states and allows the user to return to *any* of these stored states;
- single-level – allows the user to return to the previous state.

Both of these methods have been used in applications, and Microsoft-Word has employed both these approaches in its history.

Navigation

The ability to search the document and to sequentially scan through it are aspects of navigation. Scrolling allows a user to sequentially scan through a document, or go to a particular position.

3.1.3 Temporal Issues & Document Control

Time plays an important part in groupware systems, and this applies to writing systems also. The speed with which actions in a workspace are fed through to other users, and the results of actions are fed back, is important. This enables all views of the workspace to be kept synchronised, and users to form an accurate model of the cause-effect relationships. Writers may work on the document simultaneously or at different times. Different versions of the document may exist in parallel, concurrently. The way the work and the document are divided and controlled can require special consideration. Parts of the document may be confidential, so locking becomes an issue.

The following sections deal with issues relating to the temporal aspects of writer's interactions, and control of the document.

Synchronous / Asynchronous Use

A writing tool must consider both synchronous and asynchronous modes of work. We cannot assume how writers will work, or how they will manage the document. Writers must be able to disconnect from the group to work in an asynchronous manner, comforatable that their work can be incorporated later without conflict. Writers working synchronously need to be sure they are in fact seeing the same image of the document, and need an awareness of what colleagues are doing.

Version Control

Co-workers may work on parallel versions of a document, or parts of a document. Version control systems allow for the management of different, and parallel versions of documents. This is a common requirement in software engineering applications, and a collaborative version support system has been included in the SEPIA cooperative hypermedia authoring system (Haake & Haake, 1993).

Document Control

Documents may be divided into parts for a variety of reasons. There may be natural partitions, as in the case of hypermedia documents, or the division may be related to authorship, with experts writing each chapter. Control of the document segments, and access to them, are important to ensure document consistency.

Software engineering systems often employ revision control systems, such as RCS and SCCS, to formally control access to documents. Other users will require a more relaxed approach, using social protocols to dictate access.

Locking of documents can occur at a variety of levels of granularities. These range from being locking of files, through to character level locking.

3.1.4 Annotations

In reviewing the text, there is often a need to make comments about it. Arbitrary links between text in the document, documents, or any other objects present are desirable. Annotations are important for a variety of reasons, including the following:

- meta-comments – these are external to the main text, but about it;
- reference material – this may be from external sources, part of a plan etc;
- clipboard – an annotation may be used as a temporary storage area;
- shared discussion area – for argumentation;
- shared communication area – for messages, perhaps directed at another party.

Annotations may be private, broadcast to the group, or for a subgroup. The ability to link annotations to areas of the document allows commenting *about* the document to be kept externals to be constructed *about*

Annotations that can be linked to areas of the document can also provide further functionality. The ability to link areas can provide a hypermedia type environment in which the user can navigate between related documents. As the links are external to the document, annotations can be made without affecting the main document.

This idea may be extended to consider some aspects in the planning stage. Rather, in researching, ideas may be generated and stored as 'capsules' of information for later (or immediate) reference. Such items may be considered annotations without any necessary link to the underlying document.

Annotations need not be restricted to just text, but rather may include figures, diagrams, movie, or audio captions.

3.1.5 Navigation

Navigation aids are desirable in large documents. Scrollbars form a sort of spatial navigation aid, but there are also some examples that take into account the semantics of the document. MS-Word for instance provides an 'outline' mode that presents a levelled view of a document, and display headings to a certain depth. Manipulations, such as moving or deleting sections, are also possible in this mode.

Structural representations of documents can provide contextual information as well as being useful for navigation by providing meaningful cues. This information may be presented in linear format, or if appropriate as a tree or graph. The depth of these structures is also often user-configurable.

Semantic information may be able to be obtained from the document itself, and this information may provide navigational cues. There is also a possibility that information thus derived may be manipulable and change the underlying document

3.1.6 Summary of Single-User Writing Requirements

Table 3.1 summarises the rationale used for single-user writing requirements.

Issue	Relevance
Support the phases of writing	Tools for the planning and review phases tend to be dependant on the type of document being written. Such tools may be integrated into the environment, or provided in a palette of tools.
Meta writing	Writers may comment about the document rather than write in it
Support basic editing	Writing will still be a large part of the writing process
Document control	Many ways are available for controlling the document. Often best left to the group to decide
Version control	Important if concurrent parallel versions are needed. Useful in software engineering, or engineering documentation
Abstract representation	Appropriate representation eases cognitive effort. For example, having the document structure present can assist navigation, and is a useful aid when writers are discussing, or manipulating, the structure

Table 3.1: Design Rationale for Writing Requirements

3.2 Collaborative Requirements

So far the design issues have dealt with the single user use of systems. Section 2.1 presented examples of some groupware systems, and related problems experienced. This section deals with the extra requirements raised by allowing collaborative use of a writing tool. The issue of workspace awareness is a central theme of this research. Section 3.2.1 investigates methods for providing workspace awareness support. The usability issues relating to collaborative use of recoverability, responsiveness, and predictability are discussed in sections 3.2.2 and 3.2.3.

3.2.1 Representing Workspace Awarenesss

The importance of workspace awareness in groupware systems was discussed in section 2.1.5. Table 3.2 shows some methods for providing workspace awareness and attempts to capture information along the 3 dimensions identified as making up workspace awareness. The "who" column

method	who	where	what	comments
line number				very low-level
telepointer	⬤ *T* ⤡	•		gesturing
teleselection	⬤ *T*	• ▭		avoids parcel-post
insertion point	⬤	•		where typing goes
multi-scrollbar	⬤ *T*	▭ ■		
gestalt view	⬤ *T*	▭ ■		
floating image	⬤ *T* ☺	▭		
fisheye lens	⬤	▭ ■	🔍	
Legend	⬤ = Colour	• = point	🔍 = magnification	
	☺ = Icon	▭ = low range,	👁 = if visible	
	⤡ = Orientation	local detail		
	T= Text	■ = global context		

Table 3.2: Methods of Workspace Awareness

presents the methods used by each technique to provide information about the owner of the widget. The "where" column shows location information that is provided by each method. Methods that do not provide global information may not provide any location information if their widget is outside of the viewport area. For example, if user *A* is working in the first third of a document, and user *B* in the final third, then user *A* will not have location information about user *B* using methods such as telepointers. The "what" column indicates how much detail about *what* a user is doing is provided. This information can only be provided if the user's location be visible in the viewport.

Users may have changing requirements as to the degree of workspace awareness they require. Users working alone on a task will need less detail than users in a close collaboration. These needs will change as co-workers enter an leave collaborations. There is a need for a low-cost technique to alter awareness, so that users can form lightweight collaborative sessions. Providing a method to easily change awareness will increase the likelihood of lightweight collaborations being formed.

Fisheye Views

Section 2.2 discussed some applications that fisheye views have been applied to, and section 2.2.1 examined their use in providing workspace awareness in some point systems. A goal of this project is to place this technique in a realistic environment to determine its effectiveness and applicability.

The fisheye lens method provides varying amounts of *what* information, based on the magnification setting. The method also provides the user with local detail and global context. Fisheye lenses have also been demonstrated in point systems as a technique (Greenberg *et al.*, 1996), with the advantage of providing global context and local content.

How the technique is incorporated into the environment will greatly impact its usability. A fisheye lens is a means to an end—that of providing workspace awareness—so an appropriate model of use must be presented to the user (Norman, 1988). The need to form lightweight collaborations means that the model offered must allow for rapid alterations to the degree of awareness offered.

3.2.2 Collaborative Undo

In section 2.4.2, recoverability was identified as a usability with special requirements in collaborative use. This section looks at the special requirements for a collaborative undo operation.

In collaborative systems the model for implementation of undo differs from the single user case (Prakash & Knister, 1992). From a given user's perspective there are two types of activity occuring in the workspace:

- ones own actions;
- the actions of others.

The *locus of control* of actions in the workspace are internal or external to the user. Undo operations could be applied globally, to the last action in the workspace, or locally to the last action by the user. Undoing the last global action is complicated because other users may still be working, and changing the 'last' action. In applying the undo locally to one's last action, more predictability can be achieved, but it is still possible for others to have taken actions that affect one's last action, rendering the undo operation invalid.

3.2.3 Speed

Activity in the workspace has to be propogated to all participants so that any differences in the workspace are extremely short-lived. This extra communication overhead will impact the speed of the system. A balance is needed between these requirements, but the Colab system Tatar *et al.* (1991) showed that providing information immediately is crucial to avoid collaborative breakdowns. This activity information needs to be provided at a fine grain in synchronous work, and may may the communication requirements to great for use over slower links such as wide-area-networks.

3.3 Supporting Evaluation

This research project consists of goals which will be achieved through experimentation. These goals are the following:

- evaluate the effectiveness of the fisheye lens technique for provision of workspace awareness;
- conduct comparative evaluation of various workspace awareness techniques.
- provide tools and support for those conducting the experiments

Section 3.3.1 investigates the support required to conduct such experiments. Section 3.3.2 considers the ethics of dealing with human subjects in experimental situations.

3.3.1 Support for Experimenters

The various workspace awareness techniques discussed in section 2.1.5 provide workspace awareness in different ways. Combinations of them may prove to be most effective for certain tasks, so the ability to selectively include techniques is desirable for experimenters. Results may be derived from observational studies; or the system could provide logging and timing facilities which could be analysed offline.

It is important that experiments are repeatable and that if subjects are required to operate under identical conditions that this can be easily achieved. Experimenters should be able to configure the experimental environment for each subject, save it, and recall this setup later.

3.3.2 Ethical Issues

In conducting experiments involving human subjects it is important to obtain from them *informed consent.* For example, at the University of Canterbury, permission must be obtained from the Human Ethics Committee before conducting experiments involving human subjects, and written permission must be obtained from them. Such permission was obtained for this project. Normally, full disclosure of what is involved in the experiment is required before commencement. This may be waived if this knowledge would affect the results, but a thorough debriefing is required in such cases.

A written transcript of how the experiment was conducted is presented in appendix A. Appendix B contains an example of the consent form used in the experiments involving DOME.

3.4 Summary

This chapter has discussed the functions and support that is needed for writers in a collaborative writing environment. The importance of workspace awareness has been investigated and support for experimenters conducting studies of the usability and effectiveness of the workspace awareness techniques. Table 3.3 summarises the rationale for the design of an ideal collaborative writing system. This rationale will be used in the next chapter as the basis for a collaborative writing system.

Issue	Relevance
Annotations	In writing a document, facilities to markup the document such as by a proofreader or editor, have been shown to be useful.
Providing details of others activities	Workspace awareness includes providing information about what is occuring in the workspace, including actions of others.
Multitasking by writers	Writers may be using two areas of the document.
Multiple focal points distinguished by use	Two reasons proposed for providing multiple foci • reference • multiple working areas
Recoverability	A forgiving interface is desirable. We provide undo facilities, bearing in mind the need to incorporate a model of collaborative usage.
Synthesis/mapping of fish-eye concept to workspace awareness	
Changing collaborative needs	When working closely, common reference is more important. (especially for deixis) When working apart there is a need for control.
Synchronous/asynchronous use	Ways of breaking up the document.

Table 3.3: Design Rationale for Ideal Collaborative Writing System

Chapter 4

DOME: A Prototype Collaborative Writing Tool

This project addresses the need to evaluate the effectiveness of fisheye lenses as a technique for providing workspace awareness in 'real' groupware systems. This chapter presents such a system: DOME; a 'real' and usable collaborative writing tool. DOME provides the functionality, and support for writers working together or alone, and is also a testbed for the evaluation of several workspace awareness techniques.

The overview of the system is presented in section 4.1. Many of the points of the design rationale will have been touched on in this section, and these are fully covered in section 4.2. Section 4.3 discusses the problems and issues that were encountered during implementation.

4.1 System Overview

DOME is a collaborative writing system that emphasises support for workspace awareness. Workspace awareness concerns factors that occur in synchronous use. Many of the features in DOME will provide support for both synchronous and asynchronous use. No explicit support for asynchronous activities, such as mailmerge operations, are provided in DOME.

DOME, shown in figure 4.1, provides the basic functionality found in most word-processing systems. This was cited as an important requirement for a collaborative writing tool in section 3.1.2. Documents can be saved to disk, and retrieved later. DOME provides a direct manipulation style interface, and objects, such as characters and words, can be selected with a mouse. These selections may be cut or copied to a local clipboard, or deleted. Selections external to DOME can also be placed into this local clipboard by using standard X-windows copy and paste mechanisms.

A single level undo offers users the ability to recover from errors, and to experiment safely. A simple search facility has been implemented, and a scrollbar allows users to slide a window over the document.

DOME allows writers to operate in a relaxed WYSIWIS

Section 4.1.1 investigates how DOME deals with issues relating to group use. Section 4.1.2 examines how workspace awareness is provided in DOME. Section 4.1.3 discusses how DOME provides support for writing.

4.1.1 Collaborative Issues

This section deals with the issues that arise when more than one person is in the workspace.

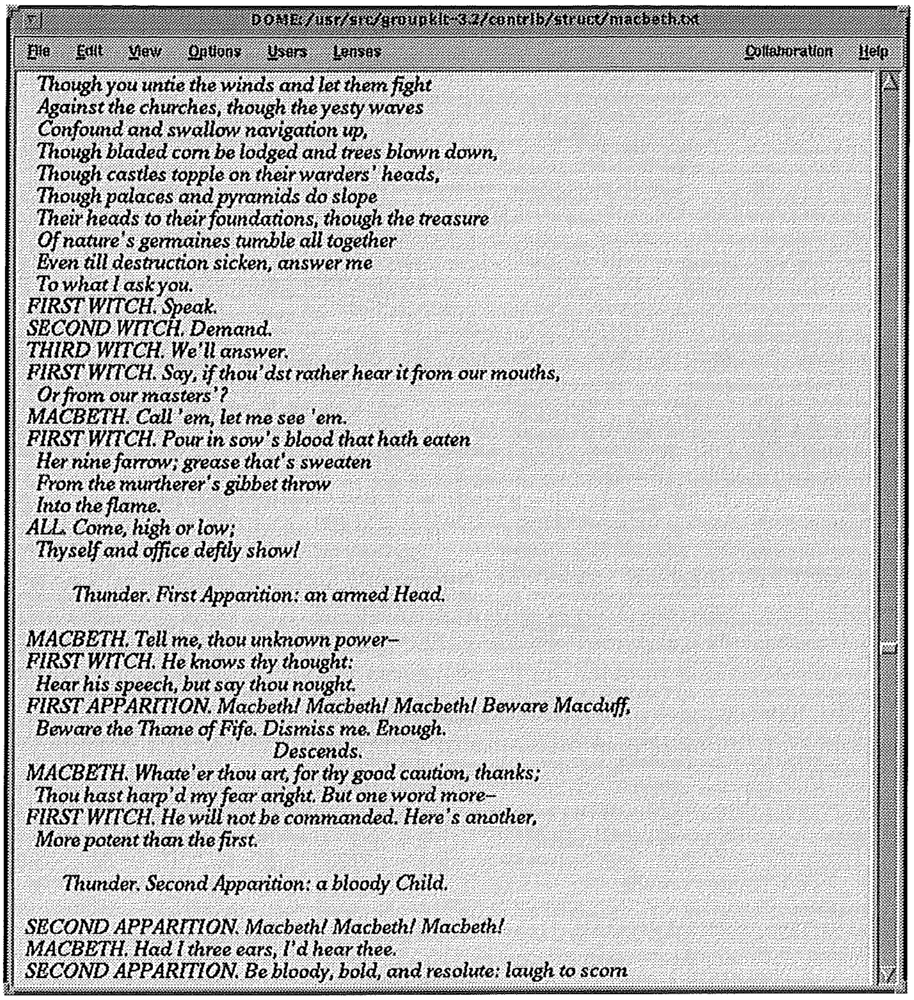


Figure 4.1: Interface to DOME

Teleselections

Teleselections provide a means of sharing information about the selections other users have made. This can be useful as a means of providing an indication of something about to happen to that selection. When a user drags their mouse to extend their selection area, that information is transmitted to other members of the conference. The area that has been selected appears on other user's screens in the user's representative colour.

Collaborative Undo

DOME implements a single level collaborative undo. In implementing a collaborative undo, one needs to keep track of the whether other users have affected one's ability to undo one's last action. For example, if user A insert a character in the seventh line, then user B deletes the first ten lines, user A should no longer be able to undo their last action. DOME is able to handle this eventuality, and many similar cases.

View Alignment

Users wishing to work in tightly-coupled collaborative sessions are able to align their views. One user will select to align their view with another, and until the alignment is cancelled will have a strict-WYSIWIS view of the other user's screen. When the leader moves their screen by scrolling, the follower's screen will be moved as well. This enables the collaborators to keep synchronised easily.

4.1.2 Support for Workspace Awareness

DOME implements many techniques for providing workspace awareness as it is also a platform for evaluating these techniques. The workspace awareness techniques provided are the following:

- telepointers;
- teleselections;
- insertion point cursor;
- user image icons;
- multi-user scrollbars;
- gestalt viewer;
- fisheye lens.

Figure 4.2: Users window

The *who* aspect of workspace awareness is provided in many of the techniques by the use of colour, as was indicated in figure 3.2. The users window, shown in figure 4.2, provides information about the users in the conference. Each user is represented by a graphic image, their name, and a colour that is used to represent them in the workspace.

The various methods are provided by selecting them from the 'View' menu, and this is discussed further in the discussion on support for experimenters in section 4.1.4.

Fisheye Lens

The fisheye lens is a workspace awareness technique that can provide detailed information about what a user is doing, as well as their identity and location. The purpose of the fisheye is to present an appropriate level of detail about a user and their activities. This technique needs to have a low-cost to access it, and be simple to of use, so that users are able to form lightweight collaborations easily and quickly.

The fisheye text viewer supplied with the Groupkit distribution (Roseman & Greenberg, 1992, 1996) requires users to configure the fisheye by manipulating 'handles'. This is a time consuming consuming process, and is not appropriate if lightweight collaborations are to be supported. DOME avoids complex lens manipulation needs by providing a set of preconfigured lens settings. , which a user can switch between, allows a user to rapidly and easily change their awareness requirements in a manner suitable for lightweight collaborations. These settings can still be configured if users require more precision in their lens configuration.

Figure 4.3 shows two users in a conference, with awareness levels corresponding to those shown in the users window of figure 4.2. Modifying the awareness levels is simply a matter of selecting a more appropriate level from the popup list available in the users window.

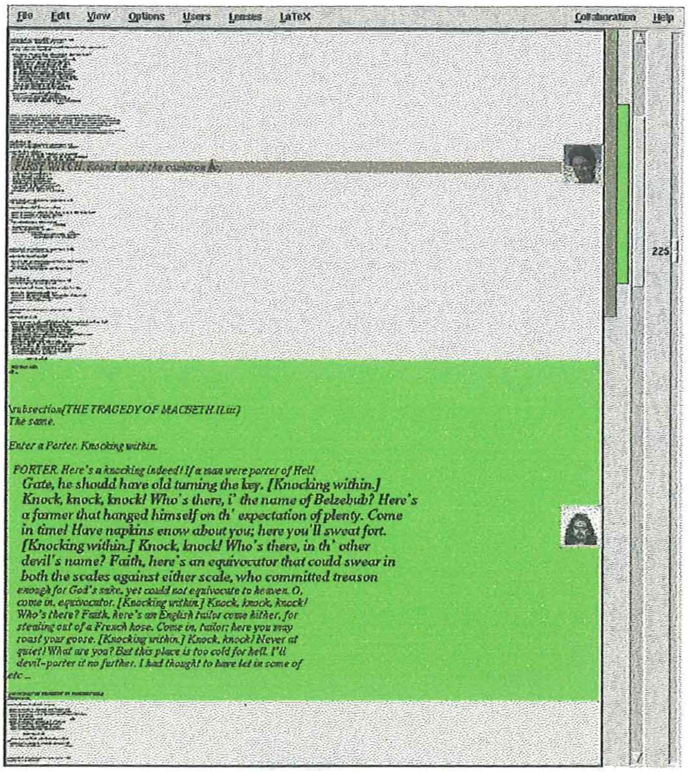


Figure 4.3: Fisheye lenses providing workspace awareness

Lens Configuration

Up to 10 levels of awareness can be configured, and each awareness level can be assigned a name. This name can be used to assist users in choosing an appropriate level of awareness easily. Figure 4.4 shows the interface for configuring a fisheye lens. This is not necessarily a 'normal' user function, as the awareness levels should suffice if they are suitably configured.

The interface shows a line indicating the centre of the fisheye lens, and handles that indicate the location of vertices of the lens. Clicking on a handle selects it, then arrow keys can be used to manipulate its position, or the delete key used to remove it. Selecting the centre line invokes a popup menu that allows handles to be created and other configuration functions. A direct manipulation interface has also been implemented, and problems with this are discussed in section 4.3.

As well as configuring the lens, the size of the background text is an important factor in the fisheye lens technique. The size of the background text determines how much of the workspace is contained in the current viewport, and whether this text is readable. The user is able to use the system in a manner similar to a folding editor (King & Leung, 1994) by hiding all background text, or as an undistorted editor by removing the fisheye and adjusting the background text to be readable.

4.1.3 DOME as an Environment for Writing

Section 2.3 discussed the need to *support* the needs of writers without forcing them into roles that may become restrictive. The important phases of the writing process – planning, translating, and review – were identified. The following sections present features designed to assist in these phases.

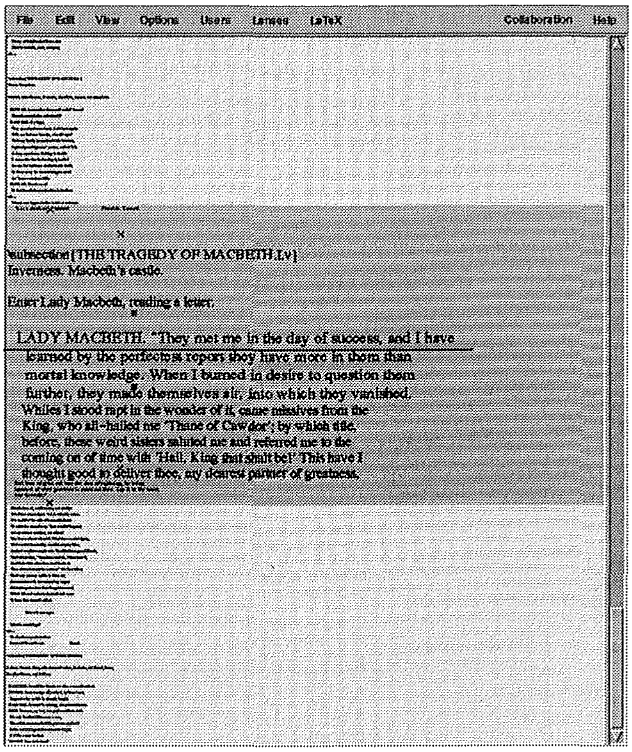


Figure 4.4: Configuring a fisheye lens

Annotations

DOME provides users with the ability to create annotations, called 'notes', which allow text to be entered in a 'post-it note' form. Figure 4.5 shows such an annotation. These notes may be linked to text in the document, and will move with the text if it is scrolled. The notes may also be independent of the document and will 'float' above it when the document is scrolled. If there

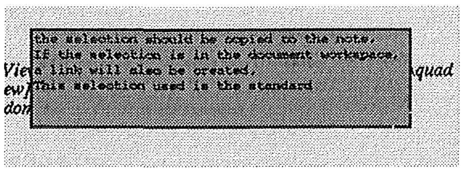


Figure 4.5: A postit note

is a current selection, either in DOME or in another application, when a note is created DOME will copy that selection into the note. If the selection is in the document, the note will be linked to the document, and the note will follow the text. Notes also have access to DOME's local clipboard. Selections can be copied from notes to the clipboard, and pasted into notes from the clipboard. Notes can be selectively hidden, or destroyed when they are no longer needed. Users can also hide or show all the notes in the workspace.

Postit notes can be created for private or public use. Public notes will appear on all windows in the conference, and may be modified by any user. Private notes appear on the creator's window only, but may be shared later by 'posting' them to specific users, or broadcasting the note to the whole group. This selective posting allows subgroups to form and have a means of communicating privately. Postit notes may be moved and resized, and this is done in a strict WYSIWIS manner. Typing into shared notes is fed through to all users who have access to the note. Notes can also provide a form of asynchronous communication – a form of email for users in the conference.

Notes currently cater for text only, but they could easily be extended to include graphics as well as text, which would enable them to be used more effectively in brainstorming sessions as drawings can allow certain concepts to be expressed more easily.

Document Structure

Section 3.1.5 discussed the importance of providing document structure, both for navigation and manipulation. DOME provides a structural view of documents as an aid to navigation. This

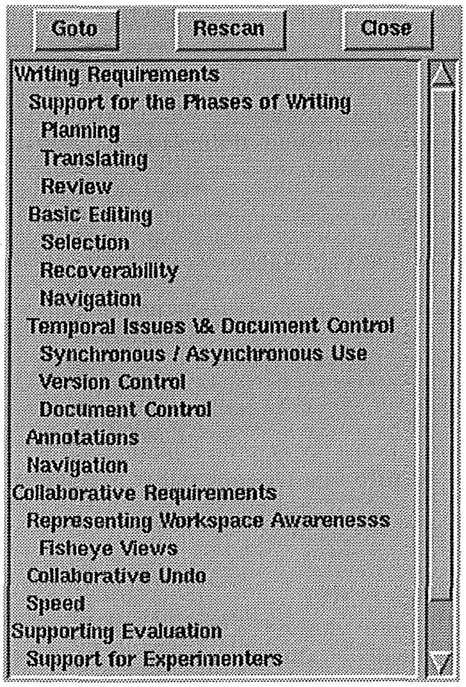


Figure 4.6: Document Structure Window

information is derived from the document, and presented in a window as is shown in figure 4.6. DOME is currently able to derive structural information from LaTeX documents and Tcl/Tk source code. A GUI version is currently being worked on, and this is discussed in section 6.1.

4.1.4 Support for Experimenters

Support for experimenters is provided by the following factors:

- The various workspace awareness techiques can be including by selecting them from a menu;
- The current environment settings can be saved, so that uniform settings can easily be used between subjects, and between experiments;
- Additional users do not require login accounts. User images are accessed based on the username entered in the Groupkit registration window, rather than based on the login name.

No timing or logging information is currently obtained from DOME, because the experiments are aimed at higher level interactions.

DOME also allows the fisheye lens to be turned off, which will be a useful feature for comparative studies.

4.2 Design Decisions

Many design decisions were presented in section 4.1. This section presents additional design decisions and rationale that have influenced DOME's implementation.

This project had a time limit, which was the major constaint preventing a full implementation of the ideal design.

The workspace awareness issues that are being investigated relate to synchronous use, so explicit support for asynchronous use has been omitted from DOME.

Table 4.1 presents design requirements for DOME.

Support close collaborations	provides a *view alignment* to implement strict WYSIWIS view between linked individuals.
Annotations	post-it notes provided. These may be private or public.
Support changing awareness requirements	Preset awareness level allow rapid changes, and enable lightweight collaborations to be formed easily
Meta comments	These implemented as postit notes
View of multiple parts of document	It was considered that these be implemented using the postit notes..
Close Collaborations	Provided *View Alignment* to implement strict WYSIWIS view linking between individuals. Indication of the presence of a link is shown in the 'Users' window by lines linking the aligned parties. Also implemented a 'looser' collaboration mechanism, allowing a user to 'goto' the same place in the document as another. This option merely changed the focal point, so at any time they could type, or use an arrow key to return to their insertion point. This is modelled loosely on the way MS-Word does this with scrollbars, but in tutoring users on MS-Word, this feature was often hard to understand.
Immediate update	Important in synchronous groupware that actions by a user passed through to others immediately, and also that intermediate results are also passed through to avoid any view disparity. Keying into the shared workspace caused updates to be passed through. Teleselections passed through as they were made. These features should also be provided in annotations Moving/resizing these annotations shared also. in close collaboration mode scrolling also done. Note: Changing window size not done.
Outline processor	We saw in section 3.1.5 the importance of providing a structure view of the document with which to work. A preliminary view-only structure view has been included. This will allow navigation. No direct manipulation yet, future work.
Who is present in the workspace?	Users window indicates who is present
Location of participants	Workspace awareness widgets
Deictic Reference	View alignment
Gesturing	Telepointers

Table 4.1: Design Rationale for DOME

4.2.1 Multiple Fisheyes For a User

In the design discussions, the need for a user to have multiple fisheyes was raised. The reasons for users needing multiple focal points were discussed, and we believe there are only two uses. These are:

1. to remain aware of activity in certain areas;

2. because focus is switching between the areas – need to copy from it, or use it as a reference.

It appeared that these requirements could be met by other mechanisms, if they proved to be adequate. The alternate technique proposed was to use the annotation mechanism. For the first item; an annotation can be placed at the spot to be monitored, and the presence of a fisheye in this area would be a warning. This requirement could also be achieved by techniques such as having a second session to carry out the monitoring, or splitting the screen. For the second item; annotations can be used for reference, and text in the annotation can be copied into the main document. An alternative to this would be the inclusion of bookmarks, which is a common feature in WWW browsers, and some text editors.

Other issues of concern with multiple fisheyes were; how to change to another fisheye, and how to indicate that a fisheye was the 'active' focus.

As part of the project is evaluating the effectiveness of fisheyes, the added complication of providing multiple fisheyes was considered inappropriate. The need to handle more than one focus greatly complicates the interface issues, and the possible effects of this on the evaluation process were considered too great.

4.3 Implementation Issues

DOME was implemented in the Tcl/Tk language (Ousterhout, 1993), using the Groupkit toolkit (Roseman & Greenberg, 1992) as a framework to provide support for group use.

Tcl/Tk is a scripting language providing support for the construction of user-interfaces easily. Tool Command Language (Tcl) is interpreted and uses a single namespace. The DOME system is contained in approximately 6000 lines of source code, which is approaching limits in terms of manageability because of the single name space.

Groupkit provided an *environment* feature which promised two useful features not available in Tcl. These were a way to structure data, and event bindings. After including these environments in some key sections, performance degraded quite substantially. An evaluation revealed that the *environment* feature was the cause, and was subsequently removed.

The Tk widget set provides good support for building quick general purpose interfaces. However the price for this general purpose use is a loss of control. The first attempt at providing a direct manipulation interface encountered difficulties because it was not possible to accurately position the text of the document in the viewport. The position of the first visible line could be specified as a percentage of its distance through the document. Because fisheye lenses are involved, lines are not of a uniform height, and changing the sizes of lines resulted in the text appearing to 'jump around'. The handle that was being manipulated to alter the fisheye lens required precise control, and this was not available.

Similar problems were encountered during the evaluation studies with the use of the fisheye, and this is discussed in chapter 5.

A non-mouse based alternative was therefore implemented; although the interface to this is not as intuitive. A user needs to know that the keys that can be used in this '*mode*' are $\uparrow$, $\rightarrow$, $\downarrow$, $\leftarrow$, and delete. It may be possible to implement the original, but time constraints mean this is left as future work.

With the inclusion of *awareness levels* rather than direct manipulation of a fisheye lens, this is less of a concern for users as the preset levels will usually suffice.

4.4 Summary

This chapter has introduced the DOME collaborative writing system. The rationale behind the design of this system has been discussed, and the support it provides for evaluation of workspace awareness techniques.

Chapter 5

Evaluation of DOME

This chapter documents a preliminary usability study that was carried out on the DOME collaborative writing system. The study was aimed at identifying large-grained usability issues.

5.1 Experimental Design

The experiments were conducted on the 3rd and 4th floors of the Computer Science department at the University of Canterbury, New Zealand. All subjects in the experiments had completed undergraduate degrees in Computer Science. Evaluation studies were conducted using pairs of subjects; these pairings were based on subject availability, and no attempt was made to control for factors such as gender or age. Six pairs of subjects took part in the evaluation study.

The equipment used was 2 colour Sun X-workstations both connected to Pukeko, a Sun Sparcstation, running the Sun Solaris operating system. The sessions were recorded on a Sony Hi-8 video camera that was positioned behind and to one side of the subjects. This camera was raised approximately 2 metres off the floor to provide a better viewing angle.

Before the experiment commenced subjects were asked to read and sign a consent form. his form is reproduced in appendix B. Subjects were frequently that they were able to discontinue the experiment at any stage. Features of the DOME interface were then demonstrated to the subjects, which they were encouraged to try for themselves. This procedure took approximately 30 minutes. After the subjects were familiar with the system and interface, they were asked to perform a pair of tasks designed to test aspects of the system. A transcript of a typical session appears in appendix A.

5.1.1 General

The evaluation focused on two main properties of the system — these were annotations and the fisheye lens. The interest in these features concerned the following:

- whether they were used;
- what they were used for;
- what they were't used for;
- if there were problems experienced in using them

Observations made regarding these are presented in table 5.1. Each of the points regarding annotations occurred once only across all pairs. There was only one observation that indicated 'real' use of the fisheye for awareness, when one subject noted an amusing comment being written by a colleague. The remaining observations and reports about the fisheye lens occured more than once; the most common report being the flickering movements of the lens.

Widget	Observation
Annotations	Used as a clipboard Used to communicate Used to record a discussion
Fisheye lens	Used for awareness when coworker wrote amusing comment Not much use when in close proximity Subject suggested need for second fisheye lens Movements of fisheye disturbed subjects Subjects configured workspace to be familiar and undistorted by removing the fisheye effect More important information appeared smaller than less important information

Table 5.1: Observed Widget use During Evaluation Study

5.2 Observations

This section deals with the observations that were recorded during the evaluation study of DOME.

5.2.1 User Interface

Use of the fisheye lens

Some users were observed to turn off the fisheye mechanism, set the background text to a readable size, and use the system as if it were their everyday editor. They were reverting to their 'normal' mode of working to solve the tasks, and avoiding the fisheye lens mechanism.

One subject did use the fisheye for the purpose of workspace awareness. Other subjects may have used the lens for this purpose, it was just the response to something humourous written by the partner, that indicated the use of the fisheye. Detecting such use is difficult unless subjects provide such feedback.

5.2.2 Collaboration

Task Division

In some pairs, subjects divided up the work prior to commencement, and worked non-collaboratively. This approach is acceptable, but it may indicate deficiencies in the tasks provided, as the aim is to evaluate workspace awareness. Some users who did divide the tasks stated they would use tools such as RCS to manage the document, perhaps indicating they were drawing on their background knowledge for solutions. All subjects were students and in their studies collaboration is generally forbidden. Most were also unfamiliar with collaborative working styles and tools.

Annotations

Annotations were observed being used collaboratively for two purposes. One subject used them as a means to record discussions on their approach to a problem. Another subject, in a different pair, was observed to use the annotations as an ongoing record of what had been achieved.

5.3 User Reports

This section deals with issues that were reported by users in the debriefing phase of the evaluation study.

5.3.1 User Interface

The lack of "importance" distortion information (termed 'a priori" information by (Furnas, 1986)) caused concern for some users. A subsection could appear inside the fisheye lens, with a more major heading appearing just outside of it. A problem occurred because the major heading was smaller, resulting in a perceptual conflict. Users expected the more major heading should be a larger size.

The second task required that users keep track of two pieces of related information. One subject reported that having two fisheyes would be a useful solution to this. In the debriefing it was suggested that annotations may be able to solve this. However, each fisheye would be an active area of the document, but annotations are at best a snapshot of the document. Once the document is modified, that snapshot is out-of-date. In the task a table had to be made consistent with some text. Annotations would soon become inconsistent with the document.

5.3.2 Collaboration

Users reported that the movements in their focal point, due to the effects of coworkers moving their fisheyes, was disturbing. They felt that their focal point should remain stationary. This effect was especially noticed when pair were working very closely together.

5.4 Summary

Some major findings resulted from the evaluation study that may have major impact on future studies.

Control of the lens is an important issue, however it appears that solutions using the current toolkits may be difficult. Some experimentation, using flatter lenses, which may be more appropriate for text anyway, could yield acceptable results. If all lenses are identical sizes, and the awareness levels just alter the extent, the flicker may disappear.

The tasks used need careful planning, and should probably go through a pre-evaluation test phase. Getting subjects to use think aloud techniques, and to discuss ideas, is quite difficult. However, these techniques are crucial to obtaining results.

Multiple fisheyes were suggested as a possible approach to one of the tasks in this study. This warrants further investigation.

Subjects were often observed reverting to using DOME like a 'normal' editing environment. One advantage of DOME is that it can be configured to behave like a 'normal' text editor, and this allows comparative studies to be conducted. A longitudinal study is an option worth considering, perhaps in a subject such as software engineering, where students often must collaborate on a project. This would give subjects longer to learn of the benefits on collaborative writing systems.

Chapter 6

Conclusions

This chapter concludes the report. Section 6.1 identifies future work that could be carried out, and section 6.2 presents a final summary of the project.

6.1 Future Work

Currently DOME's awareness facilities focus on low-level details of the text—the fisheye viewer magnifies at most one screenful of text. The extent of workspace awareness can be usefully extended though structural viewers. DOME's current structural viewer is a crude one (figure 4.6), but work has begun on a more powerful graphical representation of the document structure. Figure 6.1 shows *MuVi*, which allows direct manipulation of document sections, subsections, etc, and provides equal opportunity between DOME's text representation and the structural view.

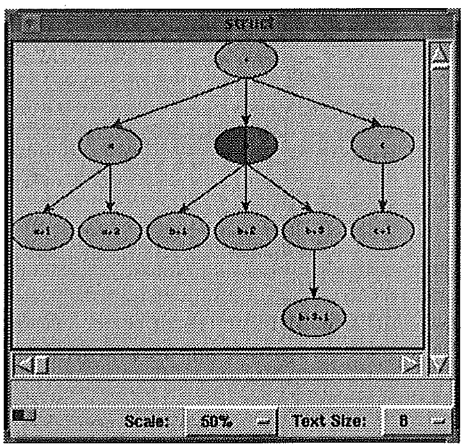


Figure 6.1: The MuVi Structure Editor

Speed is always an issue in synchronous groupware systems, and DOME is no exception. The most obvious solution would be a move to a compiled language, but this would entail replicating the functionality of Groupkit, or finding a suitable replacement.

The *annotations* could be extended to include a drawing tool, and possibly support for other embedded tools. The planning stage of writing has a need for the provision of tools, and these annotations may well provide a suitable framework for providing them.

The fisheye lens lacked the measure of *a priori* importance. This was noted in the evaluation study as a cause of confusion amongst some users. The implementation has tried to be as general purpose as possible, and it may be possible to still achieve this and provide the a priori imformation. The fisheye distortion could also be applied to other objects in the workspace, such as the annotations, and this would result in less clutter.

Further evaluation has always been the aim of this project, and the implementation of DOME has only laid the foundations for this goal. Further evaluation may take the form of longitudinal studies, where experimenters may be able to detect emergent usage patterns more readily.

6.2 Summary

Point systems have demonstrated that fisheye lenses are a powerful mechanism for providing workspace awareness. What is lacking is an evaluation of the technique in a robust, full-scale groupware application. DOME is the first such application, and incorporates extensive features for the evaluation of a variety of workspace awareness mechanisms.

A preliminary evaluation study, conducted to detect coarse grained usability problems in DOME was encouraging. The evaluation study illuminated minor functional and usability issues that should be rectified prior to full scale evaluation. Full scale evaluation, however, is the focus of further work.

Appendix A

Evaluation Task Form

This section contains the transcript used for the evaluation study for DOME.

Introduction

My name is Philip Weir, and I will be giving you instructions on what to do in this evaluation study, and will answer any questions you may have.

You are helping to investigate the effectiveness of fisheye lenses in providing workspace awareness for groups of people working together.

Workspace awareness, includes factors such as *who* else is in the workspace, *where* they are located, and *what* they are doing.

We now will look at the writing facilities provided by DOME, and later take a closer look at the mechanisms provided to assist in providing workspace awareness.

Then you will be asked to perform a pair of tasks that will give us some idea of the effectiveness of these workspace awareness mechanisms.

Before we proceed, I will ask for you to read the outline and to sign the consent form if you have not already done so.

Remember that participation in this evaluation study is voluntary. So, if at any time you feel uncomfortable, or object to anything you are asked to do, you are free to quit. We are evaluating the implementation of the software, not your performance, so we are interested in any difficulty you experience using DOME.

This exercise is being videotaped, with your consent. You may also withdraw your consent to be videotaped at any time, and have the videos erased. The video will only ever be viewed by myself and my supervisor – Andy Cockburn[1].

DOME's user interface presents the following windows:

- the workspace, i.e. the text
- images of the users currently present in the workspace.
- a view of the document structure

We will now look at these interface elements in greater detail.

[1] If you are uncomfortable in Andy viewing the video, perhaps because you are taking one of his courses, we can make provision for him to only be allowed to view the video after all related course marks are finalised for the year

Users

The Users window presents information about the participants in the session. The coloured border around a user's image is used to represent the user in the shared workspace. Other interface widgets such as telepointers, scrollbars, and the fisheye will also use this colour to represent the user.

The image is itself an active object. If you 'click' on it, a menu will popup allowing various options to be configured for the user. For instance, the colour of the user can be changed by selecting the bottom cascade menu entry.

Try this now. You will see that this colour has also been applied to your partner's view.

Beneath the image their is also a button representing the *awareness level* that has been set for that user.

Awareness Levels

If you turn your attention to the workspace, you will see your location in the workspace is indicated by the colour of the border of your image. The size of the coloured area, and the magnification effect within that area, correspond to the 'awareness level' that has been selected for you. By clicking on the button under your image you can alter this 'awareness level'.

Try this, and then try applying it to the image of your partner.

Now if you look over to your coworker's screen you will see that these settings have been applied to your view only.

The fisheye setting for each of these 'awareness levels' can be configured, but we won't be looking at that in this study.

Next we will look at how the fisheye can be manipulated within the workspace.

The Workspace

For this task if you position yourself so that both screens are visible the scrolling effect will be clearer. If one user, say the one on the right, positions their mouse cursor within the workspace window and presses <Ctrl-Button3>[2]. This binding causes the *focal point* to change and move to the current mouse cursor position. If the mouse cursor is moved while the <Ctrl-Button3> combination is retained, the fisheye can be 'scrolled' through the document.

Experiment with this until you feel comfortable with it. After both users have experimented with this, you may like to try moving through the document simultaneously.

The *insertion point* is changed by clicking in the document with the left mouse button, <Button1>. Dragging the mouse cursor with this button pressed causes a selection area to be extended.

Note that the *focal* and *insertion* points are different and any typing into the workspace will be inserted at the current *insertion point.*

We will now investigate the features offered in DOME's menus.

Menus

File Menu: This contains options to open and save files, and to exit the program, as in much other software.

Edit Menu: This contains cut, copy and paste functions which operate similarly to most popular applications.

Cut and copy operate using the current *X selection*, so one can paste from other X-windows using these procedures.

Search is not yet operational.

[2] holds down the control key and presses the right mouse button (Button-3)

Notes: these are mechanisms provided to allow annotations to be made. Writers often make 'meta-comments' about a document and this is a method provided for this. Two types of note can be created, a *Private* note to make a personal reference. Or a *Shared* note, which is generated on each user's screen.

Undo: A single-step undo has been implemented. The last operation can be undone. A *redo* of the last undo is also allowed.

View Menu: This menu is primarily designed for the experiments conducting evaluation studies into awareness. Various awareness widgets can be included by using this menu.

Options Menu: This contains some general menu options will not be used in this evaluation.

Users Menu: This contains the functions available from the popup menu in the users window.

Lenses Menu: This menu contains options to configure the fisheye lenses, and to set the size of the background text.

Select *Background Text Font* from this menu and experiment with changing the size of the background text.

"Popup" Menu: There is also a popup menu available in the workspace. This is accessed by pressing <Button3>[3], and contains choices that are available on the Edit menu.

Tasks

Now we will try some tasks. Remember that we are evaluating the implementation of the software, and we will be using some rather complex tasks. Any difficulty you experience in using DOME is important, and we are evaluating DOME, not testing you. You are free to withdraw from the task at any time.

Task 1

A production of "The Tradegy of Macbeth" by William Shakespeare is due to be performed tonight. The problem is that the script was e-mailed to the production company in parts, and assembled automatically by their automagic e-mail reassembler. This software malfunctioned and assembled the messages based on the order in which they arrived, rather than the sequence in which they were sent.

Your tasks is to work together with your partner to reassemble the text so the play can go on. The file `macbeth.tex` contains the jumbled text.

The structural representation may prove useful in this exercise, but remember that any changes to the text will need to be *rescanned* as it has not been fully integrated into DOME. These updates would slow the system too much. You may also find it useful to use multi-user scrollbars for this exercise as Macbeth is quite large.

Task 2

Your second task is to work together with your partner to co-author a paper to be presented in a "conference" for next year's fourth year computer science students.

You have been paired with someone who has done similar courses to you, or at least some in common, whilst at University.

You are asked to prepare an outline of what was involved in these courses, comments about the workload, how interesting etc.

The outline document you will use for this task also contains a table summarising across various categories.

[3]Mouse Button 3

Before launching into the task you may like to take some time to consider the various ways you and your partner could work together. Most participants will probably be university students, and Guidelines exist in most departments of the university discouraging students from working together. Because of this you may not have been involved in much collaborative writing.

The file `course.tex` contains the outline for the report.

Appendix B

Consent Forms

University of Canterbury
Department of Computer Science

Distortion Oriented Workspace Awareness in Collaborative Writing

You are invited to participate as a subject in the research project *Distortion Oriented Workspace Awareness in Collaborative Writing.*

The aim of this project is to determine the effectiveness of distortion-oriented visualisation techniques in providing workspace awareness. Workspace awareness is an awareness of what other participants are doing within the workspace being shared.

You will be working with up to two other people in completing a task involving a collaborative writing system. A typical task may be to edit a report to conform to a particular layout. During the task you will be using a variety of techniques for achieving workspace awareness.

This task will be recorded on video, and will take approximately one hour.

As a follow-up to this investigation you will be asked to make comments about your experiences in using the collaborative writing tool, and to comment on the effectiveness of each of the techniques in providing workspace awareness.

Your participation in the project should take less than ninety minutes.

After your video has been recorded you will have the opportunity to watch it. You may withdraw your participation and delete the video at any time.

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation. Only with your permission will your video be viewed by people other than the researcher and supervisor. The identity of participants will not be made public, and the videos will not be removed from the University without the participants' consent. To ensure anonymity and confidentiality, the videos and analysis data will be kept in a secured office in the Computer Science building.

The project is being carried out by Philip Weir, under the supervision of Dr Andrew Cockburn. Dr Cockburn can be contacted at 364-2774. He will be pleased to discuss any concerns you may have about participation in the project.

The project has been reviewed by the University of Canterbury Human Ethics Committee.

Consent Form

Distortion Oriented Workspace Awareness in Collaborative Writing

I have read and understood the description of the project "Distortion Oriented Workspace Awareness in Collaborative Writing." On this basis I agree to participate as a subject in the project, and I consent to publication of the results of the project with the understanding that anonymity will be preserved. I understand also that I may at any time withdraw from the project, including withdrawal of any videos and information I have provided.

Signed:

Date:

Bibliography

Ackerman, Mark S., & Starr, Brian. 1996. Social Activity Indicators for Groupware. *IEEE Computer*, **29**(6), 37–42.

Anderson, J.R. 1990. *Cognitive Psychology and its Implications.* Third edn. W.H.Freeman and Company.

Bly, SA, Harrison, SR, & Irwin, S. 1993. Media spaces: video, audio, and computing. *Communications of the ACM*, **36**(1), 28–47.

Brittan, D. 1992. Being there: The promise of multimedia communications. *MIT Technology Review*, May/June, 42–50.

Brown, John Seely. 1996. To Dream the Invisible Dream. *Communications of the ACM*, **39**(8), 30–30.

Chen, H., Houston, A., Nunamaker, J., & Yen, J. 1996. Toward Intelligent Meeting Agents. *IEEE Computer*, **29**(8), 62–70.

Churcher, N. 1995. Photi—A Fisheye View of Bubbles. *Information and Software Technology*, **37**(1), 31–37.

Cockburn, A, & Jones, S. 1995. Four Principles for Groupware Design. *Interacting with Computers: the interdisciplinary journal of human-computer interaction*, **7**(2), 195–210.

Dix, A, Finlay, J, Abowd, G, & Beale, R. 1993. *Human-Computer Interaction.* Prentice Hall.

Ellis, C.A., Gibbs, S.J., & Rein, G.L. 1991. Groupware: Some Issues and Experiences. *Communications of the ACM*, **34**(1), 38–58.

Engelbart, Douglas C. 1988. Authorship Provisions in AUGMENT. *Pages 107–126 of:* Greif, I. (ed), *Computer Supported Cooperative Work: A Book of Readings.* Morgan-Kaufmann.

Engelbart, Douglas C. 1995. Toward Augmenting the Human Intellect and Boosting our Collective IQ. *Communications of the ACM*, **38**(8), 30–33.

Fish, RS, Kraut, RE, Root, RW, & Rice, RE. 1993. Video as a technology for informal communication. *Communications of the ACM*, **36**(1), 48–61.

Flores, F, Graves, M, Hartfield, B, & Winograd, T. 1988. Computer Systems and the Design of Organisational Interaction. *ACM Transactions on Office Information Systems*, **6**(2), 153–172.

Francik, E., Rudman, S.E., Cooper, D., & Levine, S. 1991. Putting Innovation to Work: Adoption Strategies for Multimedia Communication Systems. *Communications of the ACM*, **34**(12), 53–63.

Furnas, G.W. 1986. Generalized Fisheye Views. *Pages 16–23 of: Human Factors in Computing Systems III. Proceedings of the CHI'86 conference.* Amsterdam; North Holland/ACM.

Greenberg, S., Gutwin, C., & Cockburn, A. 1995. *Sharing Fisheye Views in Relaxed-WYSIWIS Groupware Applications.* Tech. rept. 95/577/29. Department of Computer Science, University of Calgary, Calgary, CANADA.

Greenberg, S., Gutwin, C., & Cockburn, A. 1996. Using Distortion-Oriented Displays to Support Workspace Awareness. *In: Proceedings of the 1996 British Computer Society Conference on Human-Computer Interaction. Imperial College, London. 20–23 August.* Cambridge University Press.

Grudin, J. 1990. Groupware and cooperative work: problems and prospects. *In:* Laurel, B (ed), *The Art of human-computer interface design.* Addison-Wesley.

Gutwin, C., Stark, G., & Greenberg, S. 1995. Support for Workspace Awareness in Educational Groupware. *Pages 147–156 of: ACM Conference on Computer Supported Cooperative Learning (CSCL '95). Bloomington, Indiana. October 17–20, 1995.* Lawrence Erlbaum Associates, Inc.

Haake, A, & Haake, JM. 1993. Take CoVer: Exploiting Version Support in Cooperative Systems. *Pages 406–413 of: Proceedings of INTERCHI'93 Conference on Human Factors in Computing Systems* Amsterdam, April 24–29 1993. Addison-Wesley.

Hollan, J, & Stornetta, S. 1992. Beyond being there. *Pages 119–125 of: Proceedings of CHI'92 Conference on Human Factors in Computing Systems* Monterey, May 3–7 1992. Addison-Wesley.

Ishii, H, & Kobayashi, M. 1992. ClearBoard: A seamless medium for shared drawing and conversation with eye contact. *Pages 525–532 of: Proceedings of CHI'92 Conference on Human Factors in Computing Systems* Monterey, May 3–7 1992. Addison-Wesley.

Jones, S. 1992. Designing computer systems to support collaborating authors. *Pages 134–141 of:* Rees, MJ, & Iannella, R (eds), *Interface Technology: Advancing Human-Computer Communication. Proceedings of OZCHI 1992. CHISIG Annual Conference.* Bond University Australia, Queensland. November 26–27.

Kapor, M. 1991. A Software Design Manifesto: Time for a Change. *Dr. Dobb's Journal,* January, 62–68.

King, RJ, & Leung, YK. 1994. Designing a user interface for folding editors to support collaborative work. *Pages 369–381 of:* Cockton, G, Draper, SW, & Wier, GRS (eds), *People and Computers IX.* Cambridge University Press.

Kraut, R, Egido, C, & Galegher, J. 1988. Patterns of Contact and Communication in Scientific Research and Collaboration. *Pages 1–12 of: Proceedings of the Second Conference on Computer Supported Cooperative Work,* September 26–28 1988. Portland, Oregon.

Leland, MDP, Fish, RS, & Kraut, RE. 1988. Collaborative Document Production Using Quilt. *Pages 206–215 of: Proceedings of the Second Conference on Computer Supported Cooperative Work,* September 26–28 1988. Portland, Oregon. ACM Press.

Leung, Y.K., & Apperley, M.D. 1994. A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Transactions on Computer-Human Interaction,* **1**(2), 126–160.

MacKinlay, J.D., Robertson, G.G., & Card, S.K. 1991. The Perspective Wall: Detail and Context Smoothly Integrated. *Pages 173–179 of: Proceedings of CHI'91 Conference on Human Factors in Computing Systems* New Orleans, May 1991.

Malone, TW, Grant, KR, Lai, K-Y, Rao, R, & Rosenblitt, D. 1988. The Information Lens: An Intelligent System for Information Sharing and Coordination. 65–88.

Mantei, M. 1989. Observation of Executives Using a Computer Supported Meeting Environment. *Decision Support Systems,* **5**, 153–166.

McCarthy, J.C., & Monk, A.F. 1994. Channels, Conversation, Cooperation, and Relevance: All You Wanted to Know About Communication but were Afraid to Ask. *Collaborative computing*, **1**(1), 35–60.

Neuwirth, C.M., Kaufer, D.S., Chandhok, R., & Morris, J.H. 1990. Issues in the Design of Computer Support for Co-authoring and Commenting. *Pages 537–549 of: Proceedings of the Third Conference on Computer Supported Cooperative Work,* October 7–10 1990. Los Angeles.

Nielsen, J. 1993. *Usability Engineering.* Academic Press.

Norman, DA. 1988. *The Psychology of Everyday Things.* Basic Books.

Ousterhout, J.K. 1993. *An Introduction to Tcl and Tk.* Addison-Wesley.

Posner, I.R., & Baecker, R.M. 1993. How People Write Together. *Pages 239–250 of:* Baecker, R.M. (ed), *Groupware and Computer-Supported Cooperative Work.* Morgan-Kaufmann.

Prakash, A., & Knister, M.J. 1992. Undoing Actions in Collaborative Work. *Pages 273–280 of: Proceedings of the 1992 ACM Conference on Computer Supported Cooperative Work,* October 31 to November 4 1992. Toronto, Canada.

Roseman, M., & Greenberg, S. 1992. GroupKit: A Groupware Toolkit for Building Real-Time Conferencing Applications. *Pages 43–50 of: Proceedings of the 1992 ACM Conference on Computer Supported Cooperative Work,* October 31 to November 4 1992. Toronto, Canada.

Roseman, M., & Greenberg, S. 1996. Building Real Time Groupware with GroupKit, A Groupware Toolkit. *ACM Transactions on Computer-Human Interaction*, In Press.

Sarkar, M., & Brown, M.H. 1994. Graphical Fisheye Views. *Communications of the ACM*, **37**(12), 73–84.

Schaffer, D, Zuo, Z, Greenberg, S, Bartram, L, Dill, J, Dubs, S, & Roseman, M. 1996. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. *ACM Transactions on Computer Human Interaction,* **In Press**.

Shneiderman, B. 1987. Direct Manipulation : A Step Beyond Programming Languages (excerpt). *Pages 461–467 of:* Baecker, RM, & Buxton, WAS (eds), *Readings in Human-Computer Interaction: A Multidisciplinary Approach.* Morgan Kaufmann.

Sommerville, I. 1992. *Software Engineering.* Fourth edn. Addison-Wesley.

Stefik, M, Bobrow, DG, Foster, G, Lanning, S, & Tatar, D. 1987. WYSIWIS Revised: Early Experiences with Multiuser Interfaces. *ACM Transactions on Office Information Systems*, **5**(2), 147–167.

Tatar, DG, Foster, G, & Bobrow, DG. 1991. Design for Conversation: Lessons from Cognoter. *International Journal of Man-Machine Studies*, **34**(2), 185–209.

Thimbleby, H. 1990. *User Interface Design.* ACM Press, Addison-Wesley.

Thüring, M., Hannemann, J., & Haake, J.M. 1995. Hypermedia and Cognition: Designing for Comprehension. *Communications of the ACM,* **38**(8), 57–66.

Williams, G. 1983. The Lisa Computer System. *BYTE Magazine*, February.

Winograd, T. 1995. From Programming Environments to Environments for Designing. *Communications of the ACM,* **38**(6), 65–74.

Yakemovic, KCB, & Conklin, EJ. 1990. Report on a Development Project Use of an Issue-Based Information System. *Pages 105–118 of: Proceedings of the Third Conference on Computer Supported Cooperative Work,* October 7–10 1990. Los Angeles.