COMIC
ESPRIT BASIC RESEARCH PROJECT 6225

Requirements and Metaphors of Shared Interaction

Abstract

This document is COMIC Deliverable D4.1 "Requirements and metaphors of shared interaction". It presents results from the first year of COMIC on the topic of novel mechanisms of group interaction. These results consider interaction with shared objects and the modelling of spatial action around these objects.

Document ID	D4.1
Status	Accepted
Type	Deliverable
Version	2.0
Date	October 4, 1993
Editors	S. Benford (Nottingham) and J. Mariani (Lancaster)
Task	4.1

Project coordinator:

Tom Rodden
Computing Department
University of Lancaster
Lancaster LA1 4YR
United Kingdom
Phone: (+44) 524 593 823
Fax: (+44) 524 593 608
Email: tom@comp.lancs.ac.uk

The COMIC project comprises the following institutions:

Gesellschaft für Mathematik und Datenverarbeitung (GMD), Bonn, Germany
Risø National Laboratory, Roskilde, Denmark
Royal Institute of Technology (KTH), Stockholm, Sweden
Swedish Institute for Computer Science (SICS), Stockholm, Sweden
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
University of Amsterdam, Amsterdam, The Netherlands
University of Lancaster, Lancaster, United Kingdom (Coordinating Partner)
University of Manchester, Manchester, United Kingdom
University of Milano, Milano, Italy
University of Nottingham, Nottingham, United Kingdom
University of Oulu, Oulu, Finland

Editors of this report:

Steve Benford
Department of Computer Science
University of Nottingham
Nottingham NG7 2RD
United Kingdom
Phone: (+44) 602 514 203
Fax: (+44) 602 514 225/6
Email: sdb@cs.nott.ac.uk

John Mariani
Computing Department
University of Lancaster
Lancaster LA1 4YR
United Kingdom
Phone: (+44) 524 593797
Fax: (+44) 524 593 608
Email: jam@comp.lancs.ac.uk

ISBN 0-901800-31-7
Lancaster University, 1993
This report is available via anonymous FTP from ftp.comp.lancs.ac.uk.

Table of Contents

Introduction

Part I - A Spatial Model of Interaction

Part II - Shared Artefacts

Introduction

Steve Benford — University of Nottingham
John Mariani — University of Lancster

This introduction briefly summarises the major results of the strand four work, explains progress made in relation to the work-plan detailed in the technical annexe, suggests directions for further work and relates this work to that of other strands.

This document is COMIC deliverable D 4.1 “Requirements and Metaphors of Interaction”. It presents some of the results from year one of the COMIC project, an ESPRIT Basic Research Action investigating configurable mechanisms of interaction for Computer Supported Cooperative Work (CSCW). COMIC is a broad ranging multi-disciplinary project spanning Computer Science, Cognitive Science and Social Science. Overall, COMIC aims to address a range of issues including organisational support for cooperative work, CSCW design methodologies, notations of group interaction and developing novel models of group interaction. The unifying theme running through each of these is the notion of supporting large-scale cooperative work, that is CSCW systems that are populated with many people.

Strand 4 of COMIC is focusing on the development of novel metaphors and models of group interaction for CSCW. Up to now, this has involved two major and inter-related directions of research: interaction within virtual computer spaces and interaction through shared objects. The former considers the role of space in supporting cooperative work, using this as a basis for developing novel interaction mechanisms for spatially oriented CSCW systems. The latter is concerned with analysing the way in which people work together through shared information, using this as a basis for developing new CSCW oriented information storage and management services.

Part I of the deliverable focuses on the spatial work including

- review of relevant literature and analysis of current multi-user interfaces;
- identifying requirements for spatial models
- specifying a core spatial model of interaction;
- applications of this model, including “Populated Information Terrains”;
- relation of the model to previous studies of awareness in cooperative work;
- distributed systems support for the spatial model;
- observations on applying metaphors to the design of CSCW systems.

Part II describes the shared object work, including:

- results from empirical studies of the use of shared material;
- requirements of a CSCW shared object service;
- an awareness model for shared objects,

- an example application of the shared object service,
- a lightweight activity model based on the shared object server,
- the relationship between the activity model and interaction in virtual space.

The remainder of this introduction summarises the major results emerging from this work and describes year one progress, particularly in relation to the technical annexe. It also outlines the deliverable structure and touches on future work as well as the relation of the strand four work to that of other strands.

1. Summary of major results

The results of strand four are presented in detail in subsequent chapters. This section provides a quick overview of our key findings.

Interaction in virtual space

The spatial work has explored the role of space in supporting cooperative work. This approach recognises that humans are inherently spatial creatures and have developed powerful spatial skills allowing us to make sense of the world. Cognitive spatial skills such as the ability to spatially classify information and to navigate spatial environments have underpinned much previous work on user interface design, including the use of desktop and rooms metaphors. In contrast, the COMIC work focuses more on *social* spatial skills which support us in managing our interactions with other people. In essence, space provides a highly dynamic and flexible backdrop for cooperative work which encourages the development of social conventions for managing interaction. The flexibility of these conventions can be directly contrasted with the rigidity of control mechanisms in existing CSCW systems (e.g. floor control in shared workspaces and conferencing systems). As an example, consider the way in which space supports conversation management. Within a conversing group, people use position, orientation, gaze direction and gesture to signify attention and to control turn-taking. Joining and leaving of conversations is also supported through the process of a gradual approach through space. As a second example, access to shared resources may be negotiated through spatial behaviours such as queuing, jostling and scrumming.

The first results to emerge from the COMIC spatial work involved establishing a better understanding of the potential use of spatial metaphors in CSCW. This included a review of previous uses of spatial metaphors, particularly the rooms metaphor, and also an analysis of the social organisation of space. The major finding of the literature review is the need to extend the rooms metaphor to support larger scale systems. This includes extending the notion of a space within a room as being somewhere where large numbers of people can actually move around and interact and also the need to support navigation within, and “legibility” of, large structured environments of connected rooms. The analysis of the social organisation of space pointed towards the key role of *awareness* in cooperative work and of boundaries in structuring spatial environments and in enabling different modes of awareness to occur alongside different modes of interaction.

The major result emerging from the year one spatial work has been the development of a core "spatial model of interaction". The model provides a base set of abstractions for managing interactions in a wide range of spatial systems. More specifically, the model aims to maintain individual autonomy of action and also a power balance between "speakers" and "listeners" at any given time. The spatial model is defined by the following nine key abstractions:

Space — the model requires a definition of space. Space may have any number of dimensions provided that there is a consistent metric for measuring the relative positions and orientations of objects. Thus, the model might be applied to a range of systems including 2-D shared work-surfaces and 3-D virtual reality.

Objects — space is populated with objects which might represent humans, information or other artefacts.

Media — objects interact through a variety of media. Where the objects are humans, these might represent visual, aural and textual communication channels. Where they are not, they might represent more object specific interfaces.

Aura — is intended to address the problem of computational overload which arises in densely populated spaces. As the number of objects in a space increases, so the computational overhead of transmitting all changes or utterances to all other objects becomes unmanageable. An aura is a sub-space around an object which effectively bounds its presence in space and limits this problem. As objects move across space, they carry their auras with them. Only when auras collide do the objects involved establish the potential to interact with each other. How this is achieved depends on the underlying distributed system, but may typically involve exchange of references or addresses or establishment of network associations. Note that aura is a system level concept introduced to scope the number of other objects affected by the actions of a given object. Aura can be thought of as an enabler of interaction in spatial systems.

Awareness — once the potential for interaction has been established via aura, the objects themselves control the interaction that occurs via awareness. In our model, awareness is quantifiable. In other words, the system is capable of calculating how much one object is aware of another at any given moment in time. Levels of awareness determine how much as object perceives of others. For example, the level of detail seen over a visual media, the volume of an audio medium or whether or not messages are received across a text medium may all be driven by levels of awareness. Objects influence their awareness of each other by two further concepts — focus and nimbus.

Focus — is a subspace within which an object is directing its attention. The more an object is within your focus, the more aware you are of it.

Nimbus — is a subspace within which an object projects its presence or activity. The more an object is within your nimbus the more it is aware of you.

Between them, focus and nimbus contribute to awareness. More specifically, the awareness that object A has of object B is some function of A's focus on B and B's nimbus on A. Notice how the introduction of both focus and nimbus allow both the "hearer" and "speaker" of a given utterance to influence how it is managed. Thus, the interaction between the two becomes a matter of negotiation. Notice also that B's awareness of A depends on B's focus and A's nimbus and so need not be symmetrical to A's awareness of B.

Adapters — provide the model with a degree of extendibility. An adapter can be used by an object to change its aura, focus or nimbus and so alter the balance of power in an interaction. These changes may involve both amplification and attenuation. As an example, we can conceive of a "virtual podium" adapter which amplifies aura and nimbus, thereby allowing an object to make others more aware of its actions. Alternatively, a "virtual table" might fold together the foci and nimbi of its users into a common semi-private subspace (resulting in heightened awareness of events at the table and reduced awareness of those away from it).

Boundaries — are specialised adapters which serve to structure space and to control the awareness provided by the space. More specifically, as well as controlling movement, boundaries may be permeable, impermeable or conditionally permeable to aura, focus and nimbus and hence awareness.

In general then, the model is one of objects moving in space, using aura, focus and nimbus to establish and then control interactions. In general, manipulation of these concepts will be implicit, being bound up with common spatial actions such as movement. However, some explicit manipulation may also be desirable (e.g. manipulating a focal length). They will also need to be conveyed in others in natural ways using appropriate body images.

We also need to stress the generality of these concepts. Space may consist of any number of dimensions and aura, focus and nimbus may be of any shape and may be continuously valued. In fact, the best way to think of them may be as fields similar to gravitational or electro-magnetic fields in physical reality. Furthermore, in the most general case, aura, focus and nimbus may be medium specific (i.e. you may have a different set for each medium).

A further result from year one work has been an analysis of the spatial model showing how, in a much simplified case, it supports ten general awareness states such as monitoring, over-hearing, ignoring and full mutual awareness. Some of these states correspond to modes of awareness which have been identified as being crucial to successful cooperative work within real world settings such as air traffic control and control of the London Underground. This analysis provides some further justification of the potential usefulness of the spatial model.

The deliverable discusses a number of possible applications of the spatial model including multi-user virtual realities and *Populated Information Terrains* (PITS). PITS are defined to be abstract data spaces which support visualisation of, and collaboration within, shared data. Early work on PITS has identified several approaches towards constructing data spaces including statistical clustering methods, multi-dimensional visualisation, graphical approaches applied to hyper-media structures and more human centred approaches.

Year one has also involved some early prototyping work of spatial model concepts. Three demonstrators have been produced. First, DIVE (from SICS) is a fully distributed multi-user VR system which implements the concept of aura. Second, Nottingham have implemented some fly through demonstrators of nimbus, focus, adapters and boundaries using a "desktop" virtual reality platform. Third, Nottingham and Lancaster have developed an early Populated Information Terrain demonstrator.

In summary, we believe that adopting a spatial approach to CSCW promises much in terms of designing flexible and natural systems. Furthermore, we argue that the spatial model of interaction, particularly the concepts of aura, awareness, focus, nimbus, adapters and boundaries, provide a novel and powerful set of abstractions for managing interactions in a variety of large scale virtual spaces.

Interaction through shared objects

Work over the last year has examined a number of issues concerning the representation and cooperative use of shared objects. A central feature of this work is the development of a set of requirements for a shared object service which augments the facilities provided by existing infrastructures to support the cooperative sharing of information. The facilities of the shared object service identified in this document are informed from empirical results gathered from both new studies and an examination of existing widely known field studies. The main results of this part of the work are

- An analysis of a number of sociological studies considering the properties of shared real world objects in a number of cooperative settings with a view to discovering the kinds of use, manipulation and transformations that people expect to be able to carry out on these.
- The development of a set of mechanisms which challenge traditional concepts of transparency in distributed information systems. We have paid particular attention to the awareness of cooperating users and its relationship to concurrency transparency.
- A technologically and sociologically informed synthesis of a set of requirements for a shared object service which provides support for the cooperative sharing of information.
- The application of the facilities of the shared object services to support the development of a particular application.

The use of shared information in cooperative settings is not in itself unique and a number of cooperative systems have been developed which exploit some form of information sharing. However little consideration has been given to the provision supporting models and infrastructures which allow shared objects to be used in a scaleable and extendible manner. Rather an ad-hoc approach to the representation of shared information has been adopted. This is problematic in that each system concentrates on the development of a particular set of techniques which the developers wish to demonstrate or evaluate. The lack of concern of issues associated with general systems architecture allows existing infrastructures to constrain development and inhibits any integration across multiple approaches.

The emphasis of our research within this part of the COMIC project has been on establishing a set of supporting mechanisms which allow effective sharing of objects to facilitate cooperative work. To inform the technological development of shared object facilities, a number of empirical studies of work have been carried out and are included in this report. These studies have either been undertaken with a view to informing the development of facilities to support the cooperative sharing of objects or are re-examinations of previous studies. Each has regarded the use of shared objects within the physical world as been manifest in the role of paper documents and records within each studies setting.

The results of these different examinations provides a basis for the development of a shared object service which abstracts a set of facilities which support cooperative sharing. The intent of the shared object service is to provide a set of extensible and scaleable facilities which allows the cooperative sharing of objects to be managed. The separation of these facilities from the behaviour oriented facilities provided by the objects themselves allows all objects to be potentially shared in a cooperative manner. The identification of these services also allows a clear focus on the different features of cooperative interaction in a pool of shared objects. An initial set of requirements for the shared object services have been generated, informed by both the needs of real world situations that limits of shared objects. These requirements highlight the technical challenges in deriving new sharing mechanisms and the associated need to overcome the perceived drawbacks of existing technological support.

A crucial aspect of cooperative sharing is the need to overcome the existing view of strict transparency prevalent within supporting infrastructures. This in itself requires the consideration of mechanisms to support an awareness of other users across the shared object set. The development of techniques to provide this has been a central feature of the work of this strand and the realisation of these mechanisms within a shared object service is reported. To complement the development of requirements informed from cooperative work settings and to verify the construction of novel mechanisms we have considered in what way a shared object service can meet the needs of cooperative applications. In particular, we have examined how the shared object service supports the knowledge net a cooperative application. We have also considered how shared objects within the can be used as the supporting basis for cooperative activity management.

Rather than focus on the considerable technical challenges for a shared object server, it was vitally important that some form of sociological input into the aspects of sharing that need to be supported was provided. This has resulted in two documents which report on analysis of real world settings, identified what is seen as a shared object in the real world and what kind of manipulations people expect to be able to carry out on them.

An early realisation of the work was the need to overcome some of the traditional transparencies associated with distributed information sharing systems. This has initially focused on the traditional database management system mechanism for concurrency control -- transactions and locking. Concurrency control is needed to ensure the consistency of the data is never breached, by ensuring that any updates and access do not see the midway results of any other transaction. This is done by achieving a concurrent schedule which is equivalent to a sequential schedule of transactions. The net effect of these mechanisms, however, is that each user is isolated from the others, and has the illusion of being the sole current user of the system.

We believe that there is a strong need for awareness; for users to be aware of other users' presence and activities in the system, and for the need to support communication amongst users should they so require. This has lead to some prototypical work within a specific database application which demonstrates the possibilities for providing awareness and for presenting this to the user. Part of the future work for this strand will involve expanding the concept of awareness to other transparencies.

The requirements for a shared object service form a major part of the deliverable. We have examined a number of "traditional" database and object services and essentially challenged them with the concept of true sharing of objects. This has resulted in a set of focused services with detail and discussion of what is required of them. The sociological studies carried out have been used to generate a number of these "challenges" and thus to inform the evolution of the requirements. It is important to note that we are not suggesting the creation of a single COMIC shared object service developed from scratch; the project has monitored and will continue to monitor the work of groups such as OMG, etc. It is intended that the results of our requirements study will be fed into these standardisation bodies.

In order to assess the usefulness of the services, we have examined the use of the shared object service to support the development of a particular application. In this case an application called the Knowledgenet which intents to support the cooperative sharing of knowledge is mapped onto the facilities provided by the shared object service. This application also highlights the importance of the facilities provided by the shared object service in terms of the recent interest in the concept of group or organisational memory within CSCW.

Finally, we have considered the relation between the facilities to support cooperative activity provided by the shared object service and those of existing systems which aim to support planned and managed cooperation. A lightweight model to support this form of cooperation has been designed which allows members of cooperative endeavour to be aware of the state of the activity they are in. In addition the use of adapters within the shared object service allow shared objects to present themselves in a contextually sensitive manner. A prototype platform which realises portions of the activity model are presented. The lightweight model presented in this section is intended to support the spontaneous and unplanned cooperation allowed within a shared virtual space.

2. Deliverable structure

The strand four deliverable is divided into two parts. Part 1, A Spatial Model of Interaction, focuses on the spatial aspects of the strand four work. Part 2, Shared Artefacts, discusses interaction through shared objects.

PART 1 — A Spatial Model of Interaction

Chapter 1 reviews previous use of the rooms metaphor in CSCW and related areas, including user interface design, virtual meeting rooms, virtual reality and media-spaces. Chapter 2 then justifies the spatial approach by considering the role of space in supporting cooperation in the real-world and uses this as the basis for developing key requirements for the COMIC spatial model of interaction. Chapter 3 presents a critical analysis of the use of the WYSIWIS paradigm and video as a communication medium in previous CSCW systems. This analysis highlights the issue of synchronisation in shared workspaces and introduces a framework for discussing synchronisation issues. Chapter 4 proposes a core spatial model of interaction and introduces the key concepts of aura, awareness, nimbus, focus, adapters and boundaries. It also discusses applications of the model, relates the model to previous studies of awareness in cooperative work and considers distributed systems support for the spatial model. Chapter 5 comments on the general advantages and disadvantages of metaphor in the design of CSCW. This theme is further expanded by chapter 6 which reflects on the application of the rooms metaphor to a specific system, Co-Desk, developed at KTH. Finally, chapter 7 summarises the spatial work and identifies issues for further work.

PART II — Shared Artefacts

Chapter 8 presents the results of a study into how documents are shared in a real-world setting, namely a small entrepreneurial firm. This is expanded by chapter 9 which considers the role of documents in three further cooperative work settings. This chapter ends with some high-level requirements for a COMIC Shared Object Service. Chapter 10 describes a detailed list of requirements for a COMIC Shared Object Service, focusing on the issues of locking, versioning, history, access, queries and views, events and subscription, awareness and interaction. The chapter also proposes some basic techniques by which these services might be realised including the use of object adapters to modify the behaviour of an object in a cooperative system. Chapter 11 revisits the notion of awareness, this time from the perspective of shared objects. The chapter introduces an awareness model where users are made aware of different events (e.g. other people reading, modifying or deleting data). Furthermore, this awareness is modified by the "distance" between objects being accessed. Chapter 12 describes a specific example of a Shared Object Service, the Knowledge Net, a system for establishing, sharing and maintaining group knowledge. The chapter discusses how the Knowledge Net deals with key issues from chapter 10, namely events, awareness, history and queries and views. Chapter 13 extends the notion of events within a Shared Object Service towards the definition of a "lightweight" activity model. The aim here is to show how activities can be modelled in terms of awareness of key public events generated by shared objects. Finally, chapter 14 presents some early discussions on the relationship between the lightweight activity model and interaction in virtual spaces.

3. Progress in relation to the technical annexe

Overall, we are pleased with progress on strand four. We believe that the review work has been wide ranging and thorough and that the models and metaphors generated so far are both novel and exciting. The following paragraphs summarise our progress in relation to the technical annexe work-plan. The technical annexe defined the overall goal of strand four to be the following:

> "The goal of this work package is to develop scaleable metaphors and models of interaction applicable to larger groups. In contrast to the notational approach of work package 3, this work will focus upon the exploitation of shared artefacts and the use of spatial metaphors to mediate person to person interaction. One result will be a model of multi-user interaction with shared artefacts and associated user-interface techniques. Another will be a so called "spatial model" of group interaction in virtual computer spaces."

This deliverable describes results from task D4.1, requirements of multi-person interaction. The specific goal of this task was defined to be:

> "This task will develop requirements and metaphors for multi-person interaction both in terms of person to person interaction in a virtual space and with and through shared objects."

The following lists the activities that were to be carried out within D4.1 and describes how they have been tackled and where in the deliverable they are discussed.

> Examining how existing CSCW multi-user interfaces function to link small numbers of people, and whether this can be "scaled up" to larger, or indeterminately bounded groups including how media and interfaces currently used to link large numbers of people can be exploited.

> Examining how people interact through operations on "shared material", including editors, drawing surfaces and modelling and management tools.

The results of this activity are presented in chapter three which considers the key problem of synchronisation in current multi-user interfaces and shared workspaces and in particular, reconsiders the notion of WYSIWIS and the use of video in CSCW.

> Review of literature from a number of perspectives on the social organisation of spatial phenomena, emerging work on Virtual Reality and Cyberspace and evaluation of existing computer systems which employ spatial metaphors.

This literature review is covered by chapter one and the early part of chapter two. Evaluation of previous systems is given by chapter one. A more detailed evaluation of a specific system employing the rooms metaphor is also given in chapter six.

> Empirical investigation of shared "real world objects" that impact on organisation members in different ways, and that can be modelled by a computer. For example, patient information and control systems objects.

The results of these investigations are presented in chapters eight and nine which consider how people shared real-world objects in a variety of work settings including an entrepreneurial firm, the police and social work.

> Empirical investigation and identification of key requirements of person to person interaction in virtual spaces. These will include issues such as navigation, awareness of nearby events, movement, formation of groups, spontaneous and un-planned interaction, and the relationship between personal and shared space. The aim of these investigations will be the articulation of specific spatial metaphors related to these requirements.

The investigation involved videoing and informally analysing social interaction within a number of real world social settings. The results of this work have made input into chapter two. We have also related our emerging model to key observations from previous CSCW studies (see chapter four).

> Development of a set of requirements and metaphors suitable for computer implementation of shared objects and appropriate interfaces for cooperating users.

The requirements are presented in chapter two. The metaphors are introduced in chapter four.

4. Further Work

There are several directions for further work. First, the next phase of strand four will involve detailed modelling and prototyping work. There are many immediate issues to be addressed.

The spatial work needs to consider issues of group aura, nimbus and focus and their relation to those of individuals. It also needs to explore support for awareness of different kinds of action as well as awareness of presence. The appropriate embodiment of users within virtual space represents another key direction for research. Beyond this, we need to consider the structure of larger scale virtual environments and associated issues such as legibility and navigation. The spatial model work also points towards the need for more fundamental modelling techniques for spatial systems such as the use of field theories. Indeed, current discussions are focusing on the notion of “fields of influence” which might model attraction and repulsion between different users interest spaces. Of course, current and future spatial model concepts require prototyping in order to demonstrate technical feasibility. In particular, prototyping of various Populated Information Terrains will become a priority.

The current work on the shared object service considers the means by which objects can present themsleves in a manner which is sentive to their context of use. This work needs to be reconsider in the light of practical experience of use. We plan over the forthcoming year to re-examiner our impirical work in light of our experiences and to prototype portions of the service. In addition, we wish to consider more fully the facilites needed from the shared interface service and the means by which this relates to the object service. Finally, we intent to test our concepts by porting a number of key applications to work on our prototype service as a means of examining the relationship between applications and service facilities. We also wish to further examine the means by which shared objects can be used to model cooperative activities by considering the relationship between our model of activity and that of others.

Further work also needs to address the relationship between the spatial and shared object approaches. There are several possibilities here. First, both aspects of the strand four work have developed computational notions of “awareness”, one being a spatial awareness of presence and the other being an awareness of actions on shared objects. What is the relationship between the two and can they be combined into a common model? Second, we need to relate communication through shared objects to that taking place across the space in which these objects are located. Several previous researchers have drawn a similar distinction using terms such as “double level language” or “explicit vs. implicit communication”.

Another point at which the two sides of the strand four work come together is in the concept of Populated Information Terrains. Here we take a large collection of shared objects and arrange them into a spatial structure. Consequently, cooperative work within this structure can be supported by a combination of Shared Object service and spatial model facilities.

In summary, there are many immediate issues to be addressed by both sides of the strand four work. In addition, we also need to explore how the two approaches relate to each other, particularly with respect to the concepts of awareness and Populated Information Terrains.

5. Relation to other strands

This section considers the relationship between the strand four work and that of other strands.

One of the key concepts to emerge in the strand four work is that of boundaries. Boundaries allow the structuring of space to situate different activities, provide different modes of awareness and to mark territory. We see a strong relationship to the notion of organisational boundaries and hence the work of strand one. On one hand, models of organisational context might represent spatial structures. This leads to the notion of a “virtual organisation” where a model is maintained of a set of virtual spatial structures which support cooperative work (the organisational model might also represent physical structures such as rooms, offices and buildings, but this is a different matter). On the other hand, the spatial model could be used to directly visualise organisation context leading the idea of a virtual organisation browser — a specialist Populated Information Terrain.

It is also interesting to consider the relationship between strands three and four. On the one hand, the spatial approach of strand four directly contrast with the more task/activity oriented approach of strand three. On the other, the light-weight activity model proposed in the shared object work presents a notation for describing key aspects of cooperative activities. How does this approach relate to the notions considered in strand three?

6. Meetings

Strand four have held the following meetings during year one of COMIC.
September 1992, Ambleside, UK (full COMIC Plenary)
January 1993, Bonn, Germany (Strands 1 and 4 only)
April 1993, Barcelona, Spain (full COMIC plenary)
June 1993, Nottingham, UK (strand 4 only)
September 1993, Milan, Italy, (full COMIC plenary)

Part 1
A Spatial Model of Interaction

1. The Rooms Metaphor in CSCW

Robert Ingram

University of Nottingham

This first chapter is a review of a cross section of systems and models which employ a rooms metaphor in their design. It aims to show the way in which this metaphor may be used across a wide variety of application areas and will evaluate each of the systems with respect to a number of defined criteria and especially for their use in a CSCW environment. The review will the highlight a number of areas which have not yet been thoroughly researched and which may form the basis of future work in the area of spatial metaphors for interaction. An appendix at the end of this report will also review some systems from the closely related area of Media Space research.

1.1. Introduction

This first chapter will be a critical review of papers which describe systems which use a rooms metaphor in some way. It will begin by stating the goals of the review and giving a number of criteria by which the systems will be evaluated. These criteria will give special emphasis to the way in which the systems employ the metaphor for use in a CSCW environment and the review will aim to highlight areas where more research needs to be carried out for the rooms metaphor to be applied successfully to a highly multi-user environment. Each of the systems will, in turn, be described briefly and evaluated according to the criteria defined. The chapter will close with a summary of the main points of each system, some general conclusions and specific proposals for future work areas.[1]

1.2. Review of Rooms Related Papers

1.2.1. Introduction and Goals

This chapter reviews a number of papers which describe models and systems which contain some sort of 'rooms metaphor', the idea of separate virtual rooms between which the user may move. Not all of the systems reviewed will use the name 'rooms' but each will embody the principle of spatial separation of work areas. The ideas and applications contained in this section originate from a wide variety of sources with diverse uses and goals, but each has looked at a rooms metaphor to aid user learning and understanding of the system and uses the spatial

[1] This paper forms part of work towards the PhD thesis of the author

organisation of data to make it easier to navigate and find information and to make good use of screen space which is a very limited resource.

Like most metaphors the rooms idea seeks to present the user with a familiar environment, so that the experience and knowledge of that environment may be used to make the introduction to a new system easier. To do this the 'room' must contain a number of familiar features which behave in the way they would be expected to behave in real life. This will make the user feel at ease with the system and able to learn it quickly, but restricts the power of the metaphor. Adding additional features and objects which are not familiar can make the metaphor more powerful, but makes the system more difficult to learn (Smith, 1987). The 'rooms' being described here are not necessarily small focused places but may actually be quite large, containing a great number of people who can move around not only between rooms but within them.

There are two main goals for this chapter. The first of these is to provide a summary of a cross section of the systems which use this metaphor which will hopefully give an overview of the systems available and illustrate the wide variety of applications to which a rooms metaphor may be applied. The second is to evaluate these systems using a number of criteria which examine a range of facets of the systems from use of communications technology to interfaces, tailorability and provision of navigation tools. I will examine how well the systems fulfil these criteria in order to establish which of them are not widely researched and so could be a basis for future work.

An appendix at the end of the report will also include a brief summary of some related research into the use of Media Spaces to aid group working.

1.2.2. Criteria for Evaluation

This section will give details of the criteria by which the papers containing rooms metaphors will be examined. The criteria are chosen to highlight a number of different ways in which the papers relate to my work area and hopefully expose some areas of research which have not yet been fully explored and which will therefore form the main direction of my own work.

Application to a CSCW Environment

The first criterion is to look at the way in which the systems described in the papers are suited to a CSCW environment, i.e. do they allow multiple users to interact with objects and each other within the rooms, and if so how is this implemented from the system and users points of view. This is an important consideration as the creation of applications to support group working is a rapidly growing area of research and is becoming an important part of everyday workspaces. I will look at the way the systems allow multiple users to access rooms, tools and other people; the use of floor control, if any, and the resolution of conflict; and the way in which users of the systems perceive each other. Many of the systems encountered will simply not fit into this category at all as they have been designed to be strictly single user. This is not a criticism of these systems as the range of applications of the rooms metaphor is very wide and a great deal of these, if not the majority, are single user applications. Where the applications allow multiple users I will try to expose the good and bad features as I see them, relating to the topic of Computer Supported Cooperative Work.

Use of Communications

The main aim of the use of this criterion is to look at the ways in which the multi-user rooms based systems allow communication between participants. Some of the things examined will be the types of media used–are audio, video or even animation used in the system, to what degree and over what communications network and what is the policy for controlling the types of media?–the way in which communications are set up–is this easy, does it have to be done explicitly or is it a function of the room itself and when is the link between two participants established?– and how would the system handle communications between participants with different communications hardware, i.e. one has video, the other does not?

Interface Type

The type of interface used in rooms based systems is very important. The use of rooms implies a very 'spatial' feel to the system so that a 3-dimensional interface seems appropriate, but the nature of the actual system may override this as it may not be appropriate to the purpose of the application. I will look at the types of interface used in the systems described and assess their appropriateness for the rooms metaphor taking into account the nature of the underlying system. Most of the papers describe systems with a graphical interface but these range from simple windows, possibly with some graphics, through multimedia systems with video windows to full blown virtual reality interfaces with stereoscopic vision and novel input devices. Also related to this will be the way in which the interface represents people within the system both to themselves and others. This is important because for a very 'spatial' interface to be natural in a multi-user environment requires a natural representation of others and also of self to really promote a feeling of presence in the virtual space.

Methods of Mediating Interaction

Many of the systems and models described in the papers allow the user to interact with other artefacts in the world. These may include simple objects in rooms, tools and possibly other users. I will look at what artefacts the system makes available and concentrate on the way in which interaction is mediated, i.e. how the user establishes, uses and ends an interaction with an artefact. For example, 2D windowed systems may use a selection system to establish interaction, where the user clicks or double clicks on some representation of the artefact, such as an icon or button. A virtual reality system could use a similar method or something more suited to the interface, such as proximity–the interaction with an artefact is established whenever the user moves within a certain range of that object. I will look at the methods used and their appropriateness to the interface type and underlying systems.

Use of Movement and Navigation Tools

Many of the systems and models being examined in this document establish a number of rooms which are interconnected in an unconstrained way. This means that the layout of the rooms can become a very complex and arbitrary graph. Studies of hypertext systems have show that people can easily become lost or disoriented even in relatively small networks (Nielsen, 1990). I will look at the way the systems described support the user in navigating around the system of rooms, assessing the usefulness of the methods used. Also in this section I will cover the way in which the user actually moves between rooms as this could be implemented in a number of ways, for example by door icons, by selection from a list of rooms, via a map, or a number of other methods. This has to be considered in the light of the above problem as different methods of movement may cause more or less disorientation for the user.

World Design and Building

Under this criterion I will examine how the 'world' created by the complex of rooms is organised and how it is built. For example some of the systems described do not allow direct movement between rooms but require the user to exit a room to some sort of overview, and then select another room, while other allow only movement to other rooms via doors. Also, some systems allow any user to create new rooms at will while others have a pre-defined layout that is completely closed to change by the user. The layout of the rooms is also very important: is the layout created automatically or by the user/designer? does it depend on the links between rooms or, e.g., the order in which the rooms were created? does a hierarchy exist, such as rooms->buildings->cities->countries?

Many of these questions are asked for semantic reasons–how the user sees the world–but they also have other implications for the system, for example producing an overview of a large number of interconnected rooms that depends on the system of links may be very computationally expensive and will affect the performance of the system.

Tailorability and Configurability

Many modern systems, especially those that are window based, now allow the users to establish a set of preferences for the systems behaviour. This has usually applied mainly to the presentation aspects of the system but there is a movement to make many more aspects of applications tailorable by the user. I will look at the way in which the systems and models described allow, or propose to allow the users to tailor the system to their own tastes and configure it depending on the resources they have available. There could also be a number of levels to the tailoring of systems in a CSCW environment, for example some aspects of the systems presentation and behaviour may have to be defined over a group rather than for an individual, both to ensure compatibility between users and/or to enforce an organisational style. I will look at how well the systems that allow group working implement this facility, if at all.

Scalability and Distribution

One of the goals of current research is to produce a rooms based system which could scale to potentially global proportions using the networking facilities of the present and future. One of the main steps towards making this possible is to produce applications that use distributed processing methods. This will mean that the systems can be split into a number of sections and run on any machine on the network. Using this method, and by organising the layout of the global rooms network appropriately it should be possible for each site to hold its own local subset of the rooms but still almost transparently wander at will from its own rooms to any held at remote sites. I will therefore be looking at the whether the systems are based on a distributed platform and the way in which they organise the rooms and access the data that describe them.

1.3. A Review of Models and Systems Employing a Rooms Metaphor

In this section systems and models which use or describe some sort of rooms metaphor will be briefly summarised and evaluated using the criteria described above.

1.3.1. The Rooms System

The 'Rooms' system (Henderson, 1986; Card, 1987) represents a milestone for interfaces based around a rooms metaphor. The goal of Rooms is to reduce space contention in window based systems by establishing, effectively, multiple desktops. The first paper begins by introducing the ideas of space contention and some proposed solutions such as large virtual workspaces and fish-eye views. It then continues by analysing desktop use by stating the way that small screens may lead to problems of flipping through 'piles' of windows and uses an analogy with paged memory systems saying that thrashing may even occur. This is taken further in the proposal that users develop 'working sets' of windows for different tasks and that problems develop when task switching due to the necessity of closing and opening numerous windows. The developers looked at the other systems for managing screen space, especially the large virtual workspaces but found that they suffered from the spatial constraints of organising single copies of a large number of windows and decided on the use of multiple geometrically oriented linked workspaces as the spatial analogy would help with navigation problems.

The Rooms system itself is then described in detail by presenting a number of possible problems with the rooms metaphor and their proposed and implemented solutions. Rooms presents to the user a number of approximately screen sized work areas, called, not surprisingly, rooms, which contain placements–windows with additional position and presentation information. Any window may occur in more than one placement and therefore in more than one room, so sharing information between tasks should not be a problem. Each of the rooms is meant to be associated with a specific task and would contain the working set of windows for that task. Also, within the system it is possible to include whole rooms within each other. This mechanism allows, for example, the establishment of a set of windows to be present constantly by including the room that contains them in all other rooms.

The user can move between rooms using doors, which are graphical buttons, or by using an overview. To backtrack easily to the previously occupied room a back door is always present. The overview is a representation of all the rooms in miniature, so it allows selection of any of them as the destination, but which contains much of the functionality of the main views of the rooms, allowing the manipulation of placements. It is also possible to overlay this view with lines showing which rooms are connected together, but this can easily become confusing and so can be turned off. Windows can be copied between rooms using the overview or by carrying them as baggage in the normal view of the workspaces. One further aid to navigation is provided, a pop-up menu containing a list of all the room names so allowing fast movement to a specific room.

The final feature of the system that needs to be mentioned is the room descriptor file. Each room can have associated with it a file which contains a script of expressions that will be evaluated on certain events such as creating, entering or leaving a room. The reason for providing this feature is to allow advanced users to tailor and evolve the system by creating more complex effects. This allows users to add extended behaviour and appearance features and to tailor the look of each room independently.

The second of the two papers reiterates the overview of the system but concentrates mainly on the way it aids the user in switching between tasks. It defines the properties necessary for smooth task switching and the way in which the features of Rooms address these problems.

Evaluation of the Rooms System

Rooms is inherently a single user system and so in evaluating it I will concentrate on its interface and application of the rooms metaphor. The system was a major milestone in the development of rooms based systems and introduced many of the ideas that more recent applications include as standard, such as the spatially based organisation, the use of doors for movement and the overview showing all available workspaces. Rooms had two main goals which were to alleviate two problems that were discovered by studying the way in which users of windowing systems manage their space and large numbers of windows. The first problem was that the screen space was much to small when the number of windows in use increased to more than a few and the second was that a great deal of work in closing windows and opening new ones was needed when switching tasks.

The core of the rooms system is essentially a window manager that offers a number of separate desktops which a user may switch between at will. The interface therefore is that of a normal window manager, allowing all the usual interaction with the windows by usual methods as well as providing the extra functionality needed to navigate around the system of interconnected rooms. This window manager must however do more than simply allow the positioning of windows, it must remember which windows are present in each room and their positions and must make these available as soon as the user enters the room, even if this means starting a number of applications running.

The system contains three basic methods of movement and navigation. The first is the door which is simply a button leading to a named room, with one door for each of the rooms connected to the current location. This could possibly cause a problem with screen space if the number of connections is large. Also moving by this method may make it easy to get lost as it provides no spatial information as to the orientation of the rooms. To aid in the process of backtracking a back door will always be present to return to the previous room. Movement may be made more directly by using one of the other two methods, a simple pop-up menu with a list of room names or the overview. The menu allows movement by simply selecting the room name, but this does not remind the user of the contents of the room and could easily become impractical if the number of rooms becomes large. The overview gives a pictorial representation of all the rooms in the system so it will give the some spatial information to the user, but this is limited by ordering the rooms alphabetically and would serve much better if the layout depended on the connections or was at least tailorable by the user. Movement can be carried out simply by selecting an option and the room to move to. Each of the pictograms of the rooms shows outlines of the windows contained and through these allows manipulation of these windows such as movement within rooms and copying and movement between rooms. For more detail on a room each pictogram may be expanded to full screen size and manipulated. Finally the overview screen may be overlayed with lines showing which rooms are linked. This however quickly becomes a tangle of lines which is more confusing than helpful, and the choice of alphabetic ordering only increases the confusion where a more thoughtful layout algorithm would decrease it.

This therefore means that the user has no control over the map of the world. All the connections between rooms can be defined by the user so the actual conceptual layout of the world is configurable but the overview will not reflect this in any real way and so any attempts to group rooms together into logical areas will not show up in this tool. The method of including rooms within each other does open up the possibility of using some sort of nested structure to group rooms on a number of levels but it is unclear weather multiple sub-levels of room are possible and also how the overview would display this information. Ideally multiple levels of overview would help but I feel that the authors, perhaps understandably, did not foresee room inclusion to this depth and so no multi-level overview was provided. As the system is single user security methods to restrict the creation and linking of rooms are not necessary.

Tailoring of the Rooms system is achieved via the mechanism of room description files. This allows operations such as copy, move and resize to be carried out on receipt of an event such as entering a room or hiding a window. This system is designed to allow users to write quite complex scripts to create new behaviour.

Also Rooms tries to reproduce all the features of the Interlisp-D environment and so it's appearance is presumably tailorable in the same way as this system. Another definable feature of the system is the background which may be defined independently for each room and may be based on different transformations of bitmaps. This gives each room a 'wallpaper' which may help in identifying rooms and which is mirrored in the overview. Finally all modifications to the appearance of the rooms are stored automatically and reproduced when the room is next entered. This may have some drawbacks if a carefully positioned window is moved accidentally as it would not be restored to its original position when by simply leaving and re-entering the room.

1.3.2. The CRUISER System

CRUISER (Root 1988) is a communications system designed to support distributed cooperative work. One of the goals of the project was to make communicating with a remotely located co-worker by computer as easy as walking to the office next door. The actual technology of the communication is very similar to that seen in the Xerox Parc/Euro-Parc experiments with media space, i.e. one or more video cameras in a rooms and common areas and high quality audio using a computer controlled high speed network.

The CRUISER system presents an interface to this technology which (initially at least) takes the form of a two dimensional plan view of the 'space' within which the users interact. This area may be modelled on the real world but can be made more effective by avoiding problems imposed by architecture and designing the world to encourage the most interaction. To achieve this the system emphasises the use of a number of common areas such as corridors, where people may meet on purpose or accidentally, and offices which adjoin these commons. A user moving past an office door will see glimpses of the images from the real office corresponding to that room and will be able to hear snatches of the conversation, and if it interests them and the people within are known to them they may decide to interrupt and join in. Communications links with specific people may be established by selecting the destination corresponding to their office. The user will then be 'moved' there, at which point the link will be opened. The process of moving to the selected location can be carried out in one of three ways, by a jump, a planned route or a random walk. The jump mechanism will take the user directly to the destination without passing through the intervening virtual space and so does not allow for informal interaction. This problem is overcome by the two remaining methods; the planned route, which follows a pre-defined path to the destination and the random walk which is generated by the software and encourages venturing into new areas of the world.

The system also has some definitions of a persons availability, the times at which they may be interrupted and their desire for privacy. A person may be completely 'out' or uninterruptable, in which case those attempting contact will be able to leave a message. Alternatively there may be a possibility of interrupting a person at work if the reason is important enough. The level of availability and privacy required can be controlled by the owner of the room by setting an availability state or explicitly setting audio/video channels to on or off. These settings are communicated to those passing the office by overlays (blinds) on the video window and the amount of audio information heard. Thus if a person does not want to be interrupted except in an emergency there will be very little video and audio information available but the more open to interruption a person is the more information they will allow out, i.e. by opening the blind and having the 'door' open to allow more sound out. Both the audio and video channels are always symmetrical so that no person may be observed without knowing about it.

As all the links to other people in CRUISER are under computational control the paper proposes that it will be possible to compute sub-worlds based on such things as recreational interests or organisational membership. This has the purpose of encouraging social and work interactions by identifying users with common interests. It is also possible to attach links to application activities so that these may result in some sort of unplanned meeting. For example attaching a link to the printing of a document may result in the type of meeting within CRUISER that is common when collecting documents from a printer in the real world. This method could also help cooperative work more directly by establishing connections to co-workers when activating an activity that is being worked on together.

Evaluation of the CRUISER System

The CRUISER system is designed primarily to encourage feelings of proximity among non-colocated workers. The way the system sets out specifically to promote unplanned meetings and interaction is an excellent goal as this is known to greatly enhance the effectiveness of collaborative working, as the papers citations show. The paper however does not mention any support for actual planned meetings, such as some sort of calendar manager or meeting planner. It appears that the system manages informal interaction very well but in doing so may not emphasise planned interaction enough. This of course may not be as big a problem as it first sounds as the informal interaction mechanisms may well be sufficient to make organising meetings a simple task. More than this the system actually contains very few tools of its own and seems to be establishing an environment within which other CSCW tools are used. The mechanism of attaching links to applications is a limited method of integrating CRUIER and other applications which may prove quite effective. The fact that each of the users in the system are equally proximate and that the links are under computer control is, I feel, one of the most important points made in the paper. It is stated that sub groups may be formed for a number of purposes and implied that some sort of organisational knowledge base could be used to compute these groups based on a number of parameters. Whether or not such a knowledge base exists within CRUISER is not stated explicitly and its possible uses are not fully explored, however, one of the aims of the system was to make interacting with a remote colleague as easy as going to the office next door and CRUISER does appear to make such communication simple to establish as well as encouraging chance meetings. The ability to form subgroups based on a variety of parameters encourages social browsing, a user may create groups based on social interests as well as work interests. One drawback of the system as described in this 1988 paper is that it cannot cope with anything more than two way conversations, i.e. it can display only a single video source on any one workstation. This is a very limiting factor as meetings of more than one person will require all the participants to be present in one of two offices. The paper implied that this would be an area for further work to concentrate on.

The communications setup used by CRUISER is similar to the Media Space systems that are now becoming familiar (Borning, 1991; Dourish, 1992; Gaver et al., 1991; Gaver, 1992; Mantei et al., 1991), i.e. video cameras and microphones attached to a separate audio and video network under computer control. The video would ideally be displayed on a computer monitor and the sound would be 3D localised, although this was not present in the demonstrator. It was decided that all communication links in the system would be symmetrical as a method of avoiding surreptitious observation and unwanted intrusion, although this was later dismissed as being too limiting in work at EuroPARC (Gaver et al, 1991) where it was decided to rely on social convention and good will, along with the option of explicitly closing channels of communication if privacy was required. This was mainly due to the feeling that such functionality as brief one way glances were important enough to outweigh the need for strict control. The designers of CRUISER have however tried to make different levels of privacy easily available to occupants of offices and to have this communicated to other users.

The main difference between CRUISER and the other media space systems is the interface which allows access to all the systems functions such as moving around and glancing into offices. The description of a possible interface in the paper is that of a plan view of the space with the rooms being sensitive to mouse press to initiate movement. Video appears in a separate window which may be overlayed with graphics. The representation of users in the CRUISER system is essentially the video camera view into their office which is presented in this window when a connection is made. It is unclear whether users may see themselves from the view of their own camera but different levels of awareness are supported with the use of blinds over the window and short duration audio/video samples. There is no mention of how the interface allows the user to use other functions such as attaching links to applications, and there are no actual pictures of the interface in action. The map presented in the browsing interface is not described in detail and it is unclear what would happen should this map become large. There is no mention of navigation tools to allow a person or location to be found easily so problems could occur when trying to find a single user among a large number of people within the map, despite the fact that all other users are conceptually equidistant. Another interesting addition might be a method of moving between rooms via doors as a method of slowly exploring the space and browsing for other users.

The actual architecture of the system is not really explored in the paper making it difficult to asses how CRUISER would scale to a large environment. The problems noted in the previous paragraph concerning the use of the map and finding users if the number of occupants becomes large may well become a limiting factor if not addressed. The system would need to have some method of searching for the location of a single user. It may also be prudent to consider methods of partitioning the virtual world into a number of areas or domains to make browsing easier, even if this is done simply for the benefit of the user in the interface and is not actually implemented at the system level. From the paper we can only assume that the layout of the world is static and pre defined as there is no mention of methods for actual users to establish their own world. This also has implications for user tailoring of the system although it may be possible to extensively tailor some aspects by attaching links to various applications and computing various subgroups.

1.3.3. The BICC MILAN System

The Multimedia Industrial Local Area Network (MILAN) system (Condon, 1991; Leevers, 1992) was developed as part of a RACE project called DIMUN (Distributed International Communications Using Networks) which had the goal to investigate the potential uses for broadband networks in one of a kind manufacturing. The first paper (Condon, 1991) briefly describes BICC's interest in CSCW systems then goes on to look at two important issues in CSCW applications, the taxonomy, i.e. how access to the application and its resources is controlled, and ownership of information. The author proposes that systems where control is enforced by a chairperson or the computer are too restrictive and cause frustration in users and instead no control should be imposed but the communications features of the application should allow the users to resolve conflict themselves. Where ownership of information is concerned the author shows the way in which information has been traditionally owned by corporations on large mainframes or by individuals on personal computers, but more sophisticated systems are now introducing the concept of group ownership of information with membership of the group being dynamic. The paper then goes on to describe the way in which these issues are addressed in the MILAN system.

MILAN provides an environment for cooperative working based on a virtual meeting room metaphor. Each room provides a number of tools for communication and access to shared information and applications. The layout of the room is divided into a number of sections which have different tasks and implications for data sharing. The main areas are a wall, with a video window which can contain up to sixteen multiplexed video images from cameras or recorded media, and a shared whiteboard; an exit door; a table, which contains icons for public information or applications; and a briefcase which can be used to access private information. The different areas of the room therefore represent different levels of data sharing. There are also shelves which allow access to shared data that need not be on view constantly, a copier to produce copies of documents, a clock and a wastebasket. Finally a log tool may be used to note and timestamp major decisions from any meeting. One of the major decisions taken for MILAN is that it would impose no floor controls on the user at all. If two people change the state of an object simultaneously they will each see the others change and be alerted that some conflict resolution is necessary. This is demonstrated particularly in the example of the shared whiteboard.

The second paper (Leevers, 1992) gives an overview of MILAN more from the point of view of its application to manufacturing SMEs. The main features of this paper which do not occur in the above are more specific details of the networking technology upon which the system may run, some detail of training scenarios used in testing the system and the concepts of the global village and electronic open plan. Each user of MILAN will develop a personal global village. This is a set of rooms which this person has either created themselves or been invited into by others. Each user may create or delete rooms at any time and can specify who will be invited into rooms they have created. The village is displayed as an overview showing all rooms, the layout of which depends entirely on the users manual positioning.

The concept of the electronic open plan develops from the extensive use of video in MILAN. There are a number of cameras in communal areas and peoples offices. Those cameras pointing at peoples faces have migrated away to give overviews of the whole room. A number of these images together on the video wall combine to give an the user a good general awareness of what is happening in the building, which is similar to that gained in an open plan office.

Evaluation of the MILAN System

The authors of MILAN have put a lot of thought into the CSCW aspects of the system, as can be seen by the consideration for aspects of control and access to information shown in the papers. The layout of the room allows all users to see and access the shared information and the absence of floor control is a novel and may avoid frustrating restrictions imposed by the computer, although it may not seem to be a good idea at first glance. Many of the group working tools available seem to have concentrated too heavily on controlling members ability to participate in discussions. The freedom given in MILAN, along with the high quality communications, allows the normal social constraints and patterns of speech to apply in meetings without causing feelings of frustration in users who really want to speak. It also allows members of the meeting to resolve any conflict by discussion rather than merely seeing the results of what may sometimes appear to be a rather arbitrary decision by the computer itself.

There are a number of tools available in the rooms to aid in meetings. The whiteboard is an object based drawing tool which allows multiple users with a slightly relaxed form of WYSIWIS, and the logging tool can be used both to record important decisions and to playback a summary of the meeting for latecomers. The copier and fax make producing and distributing copies of documents a simple click and drag operation. As this is only a demonstrator for a proposed system these tools are relatively simple examples compared to, for example, a commercial drawing package. Also notably absent is a proper shared editor to allow users to simultaneously edit a document. Access to the rooms created in the system is by invitation only as data is copied between machines. This means that users only receive copies of the rooms they are invited to enter so potentially each persons 'village' may be different. This does however cause problems with out of date copies of information held by users who have not accessed the system recently. This requires some copying of rooms to bring them all up to date. Rooms are entered via the village screen and exited using the door, which then returns the user to the village.

The system uses a variety of network technologies for voice, video and data communications allowing good awareness of other users. The voice data for the demonstrator is transmitted using telephones not integrated microphones but as this was a present technology simulation of a multimedia workstation of the near future this was not a problem. The data was carried mainly by ethernet but ISDN lines have been used quite successfully over large distances, even for video at a few frames per second. This gives a good demonstration of the possibilities for at least limited video transmission at reasonable rates at long distances. Normally however local video is transmitted on a separate network.

The interface to the MILAN system as it stands in the demonstrator is similar to a two dimensional desktop but is overlayed on a 3D looking picture of a room which serves to distinguish the different areas of the screen. The metaphor is based on the Macintosh interface with the major features replicated using Supercard. This therefore captures many of the useful features of the Mac UI, such as the standard methods of manipulating objects, but also has some of the disadvantages. The major problem is the size of the Macintosh screen which can result in clutter at the best of times but this is accentuated by the way MILAN splits the area into different sections. There is also no real three dimensional aspect to the interface which could have helped to reduce the space contention by allowing the user to 'look around' different areas of the room rather than having to have everything visible all the time. People are represented in MILAN primarily by a text entry in a list of users in the current room and the video view into their office if one is available. The text list is completely separate from the video window so it would be difficult to identify strangers by matching their name to the video image so some sort of caption on the video window may be useful. Users are also represented by an entry in a catalogue which may be used to establish connections and contains other information that may be useful.

The 'village' of rooms created by a user in MILAN is the only way of moving between rooms. It is extremely easy to create a room but is just as simple to delete the local copy of any room, which may cause problems if one user deletes his copy of a room but then wishes to re-enter the discussion that takes place there. The village itself will probably be unique to each user and so it will be impossible for one person to give directions to another on how to reach any room. This is not a problem with a small number of rooms but could become difficult if the MILAN villages grew so large that finding any single location was not trivial, especially as the structure of the village is flat, with no divisions into sub-areas such as buildings or domains.

The distributed nature of MILAN should make it easy to scale to a large number of users provided that problems caused by inconsistent copies of data on different machines can be reduced to a minimum and the speed of networks between users is high enough. As was mentioned above navigation problems may occur if the number of rooms used by any single user becomes large but if each room corresponds to a task or project this should not be too much of a problem.

The issue of individual tailorability has not yet been addressed in the MILAN project.

1.3.4. The SICS DIVE System

The Distributed Interactive Virtual Environment (DIVE) system (Brown, 1992; Carlsson, 1993; Fahlén, 1991a,b,c; Fahlén, 1993) is a distributed, three dimensional virtual world within which multiple users may communicate and collaborate. The concept was developed by the Swedish Institute of Computer Science (SICS) as part of the Multi-G project, a research effort on distributed multimedia applications and high speed networks. DIVE uses Virtual Reality technology to give the occupants a three dimensional view of the world which can be an open terrain or a series of rooms. Other people in the system are represented as boxes with facial features to show orientation. The system can run on various platforms and the processing may be shared over a number of machines so that, for example, specialist graphics machines may be used exclusively to render the 3D views while other machines handle other processing.

SICS researchers have developed a model for mediating interaction within their system which uses a concept of aura. This is a volume around each person and tool in the system which represents the space within which they may communicate, so when two auras intersect either a person-person communications link is established or a person becomes able to use a tool, depending on whether the auras are from two people or a person and a tool. This system takes the responsibility for establishing communications links away from the rooms and introduces a need for proximity which resembles the real world and also allows the easy formation of subgroups within a room, which means that a room may be occupied by a large number of people without the communications channels becoming overloaded.

Another feature of the DIVE system is the inclusion of tools which alter the aura of users who activate them or come within their own aura. For example, all the users who enter the aura of a meeting table may have their own auras merged into that of the table, so that they may hear the conversation from everybody around the table regardless of whether their auras would have intersected normally, or a podium may extend the users aura to cover the whole room. The system also allows other tools for specific applications to be included in the system although the only one to be described in great detail has been a shared whiteboard (Ståhl, 1992).

Finally, the DIVE system allows objects to exhibit some autonomous behaviour and animation which is controlled by processes called AI's (in honour of William Gibson). These processes need not actually be artificial intelligence programs they would simply control some behaviour in the graphical object, such as moving the hands of a clock or reacting to the proximity of a person.

Evaluation of the DIVE System

The researchers at SICS have put a lot of thought into the way in which multiple users may interact not only with each other but with the tools and objects around them. The project has been oriented towards CSCW since it began and has recently begun to implement 3D tools such as a shared drawing board which provides a pen to each user which enters its aura, up to a specified limit. The DIVE system seems to implement no upper limit to the number of users within each room and has as yet no locks for doors, so all users may wander freely within the system. This is not a problem at this stage with the experimental nature of the system as the virtual worlds do not contain private spaces to which access might be restricted. Instead the research has concentrated on the development of the aura concept to mediate communication and interaction. This is a very important development which is also being examined at Nottingham (Benford, 1993b). The main advantage of the aura system is that it provides a fairly natural method of establishing communications which, being based on proximity, is similar to that of the real world. This being the case it is also much suited to a virtual reality interface. The limited range of the aura volume around the representations of people within the system means that any user can easily avoid being overwhelmed by a large number of people talking at once and in a heavily occupied room a small group can easily form a subgroup without being disturbed by the conversations of others. All users whose auras overlap, and who are therefore communicating, do so on an equal basis, no floor control is imposed except the normal social pressures we feel in any situation. This does not impose artificial restrictions and should work well provided that the audio communication if of a high enough standard (Condon, 1991).

DIVE uses virtual reality techniques to provide a simple representation of users in the same room. These self-body icons are deliberately simple to reduce processing but need to be easily distinguishable. Each user may therefore define their own body shape and use it whenever they enter a DIVE world so users who meet regularly will become familiar with each others shape. It is also possible to look down and see ones own body icon or to detach the viewpoint from the icon to look at it from further away. The attaching of the viewpoint to some visible body helps to improve the feeling of presence in the world. DIVE can also incorporate video image of real rooms, but the main communications medium used is audio which is being developed with full 3D which will give some idea of the position of the sound source within the virtual world. The use of aura means that to establish a communications link with another user means first locating them in the system and then positioning yourself so that your auras overlap. This seems simple and natural and is good for a small system but may provide problems in a larger world without some sort of tool for locating a user to avoid the situation of having to wander around large numbers of rooms looking for somebody. The system therefore assumes an audio link and no real communication can be accomplished without this unless it is through some other tool incorporated into the system. Visual information is based mainly around the three dimensional graphical representation although polygons containing video images may be placed into rooms. The 3D interface is fundamental for establishing communications and viewing the world but the drawing tool at least is designed to be used on a 2D interface too, possibly in conjunction with the 3D version and hopefully other tools will follow this example.

As has already been mentioned, the 3D interface is well suited to the rooms metaphor because of its spatial nature so that rooms may be given ‘real’ size and shape and the inhabitants can wander around seeing things with real impressions of distance and movement. The interaction devices of Virtual Reality, such as gloves and head trackers can also make interacting in the virtual space more intuitive, for example, using a glove an object can actually be grabbed and moved. DIVE provides head tracked 3D views and interaction devices such as gloves and wands to provide suitable interfaces to the world in general and individual tools, for example the whiteboard may use the wand as a drawing tool and a glove may be used to manipulate objects in the world. This sort of equipment does have drawbacks. Much of it, especially the head mounted displays, is cumbersome and uncomfortable to use and most average users will not want to use if for long periods. These displays can also cause other problems such as motion sickness and loss of balance. These problems should reduce as the technology advances but for the moment more comfortable alternatives should be made available to users of the system, such as large desktop displays.

DIVE at this point has no tools to aid the user in navigating around the world. This is not a problem with the small number of rooms that are used at this time but the system is highly distributed and therefore well suited to scaling up. This will mean more people and more rooms, so eventually the size of the world will be so large that it will be difficult to avoid getting lost, especially for novice users, and even more difficult to find an individual person. Some thought should be applied to the problem of navigating around the space if it becomes large and along with this how movement is actually carried out as something more direct than wandering through rooms may be needed in a larger space.

Another problem that comes with scaling up is the way in which the world is designed, built and organised. Large worlds may be easier to navigate around, browse and administer if they are split into a number of structured domains to provide higher levels of abstraction from the individual rooms. One feature of DIVE that may help with this is the use of portals–polygons which allow movement into a completely separate world. Also, at the lower level the lessons of city planning must be learned to make the environment pleasurable to use. As I have already been mentioned, the DIVE system has only been used with small numbers of rooms so these problems have not yet been addressed. Also, the system as yet allows very little tailorability for individual users, but it is very much a research system and so this should not necessarily be expected as yet.

1.3.5. The Multimedia Assistant

This system (Cook, 1991), which I have included under the heading Multimedia Assistant, was designed to use the techniques of object orientation to improve communication and management of information. The paper begins by describing the background and development cycle of the system. The authors foresee a shift in the emphasis of use of computers from actual computation to communication, and alongside this a growing emphasis on object-oriented methods and networked heterogeneous systems. Noting these trends they set out to produce a model for the use of object oriented technology to support unstructured communication over a number of tasks and media. The model would be used to identify important issues and missing functionality, and would be based on an assumption of very near future technology, multimedia highly connected workstations, rather than looking ahead to a time when Virtual Reality may be commonplace. The design of the system is iterative and based around a number of scenes–scenarios describing different situations within which interaction takes place. The end goal is a demonstrator of the required features for the scene, although unimplemented features were allowed to be included provided that there was a plausible proposal for their implementation.

The goals of the system are then briefly described. These are; to allow real time remote cooperative working using a WYSIWIS paradigm; to use multiple media for communication; to allow integration across different tasks to reduce the overhead of task switching and give a standard environment; to provide personal secretary services to provide support and manage incoming and outgoing communication; to provide time management facilities such as a meeting scheduler; to provide end user programming facilities for individuals to tailor the system. Following this are sections describing the development of the scenes and architecture. These describe the design process from the brainstorming of scene ideas through the writing of scenes and dissemination of the technology required to implement them, the definition of architecture ideas and eventually the architecture definition.

Following this is a more detailed description of the architecture and features of the system. The main element of the architecture is the concept of shareable persistent objects in a distributed address space, on which the rest of the system is built, and also the commitment to a direct manipulation slightly relaxed WYSIWIS interface. This is relaxed in such a way that it is not necessary to show all pointers constantly and operations are not distributed until they are complete so that, for example, rubber banding of objects in not shown on all screens. It is immediately noticeable in this section that many of the features described are very similar to those of the Rooms system (Henderson, 1986; Card, 1987). This is acknowledged by the authors who wish to apply this paradigm to remote meetings. The elements of the system which relate most closely to those of Rooms are the perspectives, views and doors. Doors are once again used for movement. Views are a similar concept to the placements of the Rooms system in that they are the editable elements that contain windows. Views are themselves contained in perspectives, which are similar to rooms except that there are always two perspectives on view, a shared perspective, called the desk, and a private perspective called the shelf. Objects may be moved between perspectives by moving them to the shelf and changing shared perspectives, which reflects the idea of baggage in rooms, or by dragging an icon through a door.

In this system each door is individually named and each is supposed to be either an entrance or an exit, not both. back doors are not provided, instead the user is required to keep important doors in the private perspective. Doors may execute a script to validate access to a room but this is the only access restriction imposed as the authors did not want to be too restrictive. The private perspective is designed to act like a room included in all others would in Rooms, to provide a standard and persistent set of views across the full range of shared perspectives. New views can be created from templates stored in a globally accessible room called the Warehouse and can be edited interactively by switching into editing mode. The paper then describes an activity model–basically an organisational knowledge base–and a naming model with different levels of naming.

The next sections of the paper describe the individual scenes in more detail, detail the implementation–in Smalltalk-80 with extensions for networking and user interface building, with taken passing floor control–tell what worked and what didn't and end with a few conclusions.

Evaluation of Multimedia Assistant

This system, as I have said, is in many ways a multi user version of the Rooms system, borrowing much of it's functionality from the Henderson & Card system. Where Rooms provided a number of single user workspaces the Multimedia assistant provides a number of multi-user areas, as well as some other functionality to provide information and a 'personal assistant' support such as managing interactions with other users. The shared screens use a relaxed version of WYSIWIS which doesn't clutter the screen with multiple cursors or rubber banding, but each user has a pointer icon which they may move. These appear to be constantly visible so may still clutter the screen if there are too many users. Ownership of views is shown by border colour and by a change of cursor when a user gains control of a view. The paper states that the cursor change was inadequate when more than two users were involved but does not give any more detail concerning border colour, specifically whether there would be simply two colours denoting owned and not owned for each user or a colour for each participant, and if the latter how each colour was related to a specific user. Also, little mention is made of the applications running within the views and what sort of access multiple users would have to a single user application.

The paper does not give as much detail as I would have liked concerning the other functions of the system such as those of the personal secretary. The scenes described show the 'assistant' managing access to workstations for absent users and even answering the telephone to provide information on their whereabouts to users identified by tones, yet little mention is made of these services in the rest of the text. One problem which has been difficult to overcome, and which applies to the systems application to a CSCW environment and to other aspects, is the fact that only two workstations were available for developing the system. This means that it has not been tested with a large number of people in a single perspective, or with multiple users collaborating within different perspectives and so many aspects of the systems performance have been extrapolated from use by two users only.

The only audio communication specifically mentioned in the paper as being used is the telephone but it is stated that high quality 3D sound would be a desirable feature of further versions. The use of video is also limited in that it could not be displayed in real time although a full video network was another desirable feature for further versions. The impression given by the paper is that the best use was made of the technology that was available and that this was enough to give an impression of what was required from the system. At the same time thought was being given to the enhanced possibilities that would be provided by more technology. The descriptions of the scenes show than one user may interrupt another to establish communications but this is not explained in great detail so it is impossible to tell if this interruption is handled entirely by the system or manually using the telephone after checking the availability of the other user. However, this may once again be an artefact of a lack of completely integrated audio in the system and could well be made explicit if such a network was provided. Once a connection has been established users may negotiate about entering perspectives and use the other facilities of the system to aid their collaboration. The way in which other media, such as video, animation and graphics, are used is not really described and some screen dumps from the system in action would have helped greatly.

Some elements of the interface have already been described, especially those concerning the WYSIWIS implementation. The interface is a direct manipulation window based system but it is difficult to establish the relationship between views and windows. The Rooms system provides placements which contain application windows but views in the Multimedia Assistant are described as being composed of sub-views cloned from a limited range of templates, making it unclear where outside applications may fit in. Again, screen dumps of the system would be useful here. The representations of objects in the system are not described in detail except that they may be made into icons so it is not clear, for example, what doors look like. This would be useful to know as we are told that users should carry around useful doors in their private perspectives, which may cause problems if the number of useful doors becomes large. The way in which people are represented seems to be very much tied in with the workstation so objects, data and the secretary functions seem to be local to the workstation rather than an abstract representation of a user.

Interaction with the objects in this system is accomplished via the direct manipulation interface but this is mediated by a token passing floor control mechanism administered by a master workstation. This means that only one user may interact with the system in each of the views and that others must request control and rely on the current controller to relinquish it. This appears to be a very limiting situation even if, as the paper suggests, users are forced to relinquish control after a certain time has elapsed.

Movement around the world seems to be accomplished only via the doors. There is no mention of anything similar to the additional navigation tools present in the Rooms system, such as the overview of rooms or the pop-up list of room names. Also, doors are completely separated from the perspectives, must be explicitly created and are individually named, so connections between rooms are simply functions of the doors a user carries in the private perspective or which have already been placed in other perspectives. This system may work well for a small number of rooms and a small number of users but if the world is large, and expanding, users will not be encouraged to explore or be able to access new regions. I believe that although this would seem to be ideal for allowing a user to define their own topology for a world it will discourage interaction on a larger scale and that a more well defined topology should be provided and that user defined links should be built on top of these.

The papers describing the system give little information on how the complex of rooms is established and organised. We do know that the approach taken was that of object orientation so we can assume that all the perspectives are persistent objects but beyond this there is little information. As is mentioned above the connections seem to be functions of the doors rather than the rooms and so are quite user specific. Also we are not told how perspectives are created or whether it is possible for normal system users to carry out this operation. Doors are created in the perspective they lead to and then moved to where they are needed but this means that somebody must take responsibility for ensuring that connections are useful and sensible, especially those to new perspectives.

The issue of tailorability is a difficult one. It would be difficult for a single user to tailor the appearance of a shared perspective as this may result in a configuration the other users dislike. The system allows the user in control to move and resize windows but tailoring beyond this point is not discussed. A scripting language similar to that of Rooms is provided to attach behaviour to doors, views and perspectives but the papers acknowledge that this was not particularly successful as it became over complex due to the way in which objects could change attributes of other components and the interactions between local and global scripts.

It is difficult to say how this system would scale. It's object oriented design would hopefully make this fairly simple but there are elements of the design that rely on master servers rather than working in a distributed way, which could cause problems. Also many elements of the system seem to have been designed without considering the possibility of a large complex of rooms. For example the need to explicitly create doors in their destination perspective and move them to locations where they may be useful could become unmanageable in a large 'world' as could the necessity of carrying around 'useful' doors back to other perspectives. Also the restrictive floor control policy could make the system very frustrating for a meeting of a large number of people. As the system was developed using only two workstations it is understandable that such issues of scale may not have become apparent but they may need to be addressed if the work is continued to larger networks.

In summary, this system seems to have a great deal of potential. The ideas of using the Rooms system as the basis of a multi user system, and of using the workstation as a personal secretary, even handling phone calls are novel and interesting. To make these effective however the designers need to consider how these will scale up to large numbers of rooms and users and to remove some of the limitations on the system such as the restrictive floor control and fiddly link creation mechanism.

1.3.6. Collaborative Desktop

The Collaborative Desktop (Marmolin et al., 1991b) is a "set of generic tools under continuous development and revision, presented as a 'desktop'." The tools are designed to be simple and generic to allow them to be combined in a number of ways for different tasks. The paper describes two sets of prototype. The first is an interface demonstrator containing little real functionality while the second is a group of tools with 'real' functionality. Before concentrating on these in detail the paper describes the formation of the Collaborative Desktop project as part of a larger CSCW environment called the Knowledge Net (Marmolin et al., 1991a) and the main features of the approach taken. These are: a tool approach, making simple generic tools which may be combined to suit particular tasks and which will work well together; a rooms metaphor, to decrease social and physical distance between team members; and the idea of the electronic hallway, the inclusion of features to encourage casual interaction. The next sections describe in detail the two prototype areas.

The first of these details the interface demonstrator. This is a mock-up built using Supercard on a Macintosh. The direct manipulation interface demonstrates the tool approach by providing two toolboxes, one each for synchronous and asynchronous tools. These are installed, deinstalled and tailored using menu options. The tools are each single separate units which may be used together as the example in the paper shows:

> One could for example have a separate conferencing system with picturephones, whiteboard etc. or one could have one picturephone and one whiteboard that the user could combine if they want.

The authors acknowledge that further empirical studies are needed to establish if this, or a more task oriented approach is more appropriate. The section continues with a description of some of the important tools demonstrated in this interface such as the telephone for control of the audio channel and a video viewer to show live views from cameras or recorded video sequences. The room metaphor is represented in the interface in two ways. The team map shows members of the users team, each in their own 'room' which contains a still picture of the team member (actually a cartoon style icon), the tools they use and their current availability for interruption, as well as a short message such as "back in 5 minutes". The map may be used in conjunction with other tools in many ways for handling communications, such as making or answering a call or accessing an answering machine. Secondly, the metaphor is used in a shared room which allows team members to gather for pre arranged meetings with open communications channels and shared tools. Finally, the electronic hallway concept is included by allowing users to define any process as a meeting process. When any of these processes is being used by more than one user a communications link will be established between them.

The next section describes the individual tools implemented under the umbrella of the 'working prototype' using Interviews and distributed using ISIS (Birman, 1985). The first tool described is the Call Manager which is activated whenever a synchronous link between two or more users is established. It displays the icons of the parties and can manage up to four calls with five participants each, allowing the users to switch between calls. The next tool is the Team Catalogue which displays a list of available users and some information such as availability, last login etc. and can be used to display more detailed information in the form of a catalogue card. The Team Map has also been implemented to contain at most six members. To handle messages an Answering Machine is used which provides the functionality of a mail system but which allows audio messages to be created and displays information concerning the status of communicants. Other tools include a shared, object based Whiteboard and a Displayer, which is a rudimentary multimedia message player.

The paper concludes with a summary that discusses some of the limitations of the prototypes, such as the need for an effective interface because of the systems complexity and also the need for empirical evaluations of the tools and interface. There is also a brief description of some features planned for future versions such as cut and paste between tools and private applications, more support for multitasking and task switching and user tailoring of the system.

Evaluation of the Collaborative Desktop

Although this system does not, at the moment, use the rooms metaphor as strongly as others reviewed I felt that the ways in which rooms are used, mainly in the team map and the shared room, have potential and may in future become much more prominent features of a fully integrated environment, and so I decided to include the prototype system in this review.

The Collaborative Desktop supports group working in a number of ways. First, it provides tools for communications using text, audio and video, although how much support for video has been implemented is unclear. The use of multiple media, especially audio, is a very important provision by a system such as this, despite the fact that it is very much a prototype. Second, it provides support tools for collaborative activities such as the shared whiteboard and the answering machine. All these tools are designed to be used together to create more powerful applications. This seems reasonable although it may prove necessary to provide some sort of manager between the tools and the user to reduce the need for constantly opening and closing numerous tools, especially when switching between tasks. Third, the system provides a simple organisational information base and a great deal of status information about each user such as their availability and tools they use. Finally, there is the support for unplanned 'meetings' by defining meeting processes. I would like to see this area expanded to provide more support for chance meetings. At the moment it is required that both users explicitly tag the same application as being a meeting process and meetings only occur when both are using this tool simultaneously but a more random and frequent method may be more useful. Floor control within multi-user tools is only mentioned briefly in the discussion of the whiteboard which locks objects when they are selected by a user.

As was said earlier, Collaborative Desktop seeks to use multiple media for communications. Although the audio links appear to be in place the video tools do not appear to have been implemented at the time the paper was released. Calls are established using the call manager tool which can manage up to four calls each with up to five participants. This seems a little limiting in the number of people who may participate in each call but is fairly flexible in allowing the users to switch between four such calls. The interface demonstrator also includes a more informal way of establishing communications links based around a meeting room with open communications channels but this does not yet seem to have been implemented among the tool prototypes and so it is impossible to guess about the number of channels that might be supported in such a room. At this stage the system does not mention problems that may occur with collaboration between users with different technological resources and seems to assume audio and text links for all users.

The interface of the implemented tools on Sun Sparcstations seems to match quite closely the style of the demonstration interface of the Macintosh using Supercard except for the obvious differences in the standard Mac and OpenLook components. In the demonstrator the tools appear in two moveable bars and are represented by quite clear icons with English names for further explanation. The tools displayed in the bars may be swapped in and out using menu options. Everything is two dimensional and direct manipulation with people represented by cartoon style icons. Some more realistic representation such as a still picture may be better to make identification easier but could cause problems with screen space as they would probably have to be larger than the icons to be clearly visible. As the emphasis on rooms is quite small in this system and meetings are generally not established in a spatial way the familiar two dimensional interface seems appropriate.

Interaction with the system, objects and other users is established using point and click methods. Communications links with other users are established in the 'real' prototypes only by the call manager, which may then be used to initiate instances of other tools such as the whiteboard. This is quite restricting in that the call manager must always be invoked to use the other tools for collaboration. In the demonstration interface is the example of the shared room where connections are established based on presence in the room itself.

The system does not contain a number of rooms around which the user may move so navigation tools in this sense are not necessary but tools are provided to locate other users and information concerning them. The team map allows browsing of the other members of the team, although this is severely restricted to only six users, and provides quite a large amount of information in a small screen space.

The team catalogue allows for a larger number of users and can be used to access the more detailed information of the catalogue card.

The system does allow users to define a shared room for meetings and to furnish this room with furniature and shared tools. However, it is unclear whether more than one of these rooms may exist simultaneously and if so how users structure the layout of the rooms and navigate between them. The overall impression is that a single room is created for a specific meeting.

The users can tailor the look of the system to some extent using the X-Windows resources and may define their own icons for their personal representation and change the tools displayed in menu-bars. This however seems to be the limit of tailorability at the moment although it is specifically mentioned as a priority for future versions.

Finally, the system seems to be quite limited in many aspects to small numbers of people. This may be because it was intended by use for small teams with similar technology, but this is not explicitly stated in the paper. The limiting, for example, of the team map to six people and the communications channels to five people seems to be overly restrictive, although the later may be an artefact of the audio technology. On the other hand the system is built on top of the ISIS distributed platform and is object oriented and so should be relatively simple to scale up if the technology allows and the reasoning behind introducing the limitations (such as the lack of screen space) are overcome in a useable way.

1.3.7. The MOCCA CSCW Environment

In this section I will describe and evaluate two papers proposing a theoretical model for a CSCW environment. The papers come from a project called MOCCA, which is a group (working group 2) of the European CO-TECH programme, which is aimed at conducting basic research into CSCW Europe-wide. The first paper (Benford, 1993a) gives a general overview of the MOCCA environment, its goals and requirements, then continues with more detail on the approach taken, which is to use a number of different models to describe aspects of the extremely complex environment, and also introduces the architecture envisioned and the possible route of implementation. The goals of the environment are to provide a range of services and facilities to manage information and resources so that CSCW applications work not in isolation but together. Six models are then described which define the environment. One of these models is the Rooms Conceptual Model which is responsible for navigation around the world and finding objects of interest including people and tools, providing a backdrop against which activities take place and encouraging the awareness of objects, people and events in the virtual world. The description of this and the other five models form the main thrust of the paper.

The second paper (Navarro, 1992) concentrates more on the need for the environment, its goals and proposed architecture. Here the need for different types transparency and a ‘real world’ metaphor are examined more closely and the architecture described in much more detail. Three managers of the environment are proposed, a domain manager which groups objects into domains and acts as a broker for requests for services from objects and other domain managers; an activity manager which allows integration across different models of cooperation by registering activities in the environment and allowing applications to register an interest in these activities; and a security manager which ensures access restrictions are enforced as per the security policy of the organisation that owns the information. Alongside these would be an information store to provide common storage services to the objects in the environment and organisational database to store information on the organisation within which the environment is set. The services would be provided as an extension to those of OSI and ODP.

Evaluation of the MOCCA Environment

The papers being evaluated here describe work being undertaken to define a model for an environment within which CSCW applications may work together sharing storage and information bases and providing services for each other. As this is not an implemented system like those being described in most other papers here, some of the criteria defined for evaluation of these systems will not apply. However, the models, especially the Rooms Conceptual Model, do consider most of the basic ideas behind the criteria, though possibly not as specifically as some other papers, so this section will still use those criteria that are relevant.

The environment proposed is defined specifically for CSCW though not in the same way as some of the individual applications examined in this document. MOCCA attempts to use a wider perspective and examine the way in which services and management may be provided to link not only users working in a single application but users, applications, objects, storage and information into a single environment where all may work together. This is a large and difficult goal so the MOCCA approach of defining six different viewpoints is very good for breaking the problem down into more manageable units but wide-scale adoption of the environment would be necessary for it to be useful.

The rooms proposed in the conceptual model provide an open communications ether. The paper does not go so far as to propose the media that should be used here so we must assume that this is an issue that will be dealt with in the future, but the importance of using the best medium possible while not excluding users without the appropriate technology must not be underestimated. The good point is that the communications channel is by default completely open in every room, and further restrictive controls must be specifically built on top of this where desired. This therefore allows for easy informal communication anywhere in the system. Another big advantage of the model is its commitment to providing awareness of the presence and actions of other users in the system. This, we feel, is important in a cooperative multi user environment.

The papers reviewed here mention nothing explicitly about the type of interface to be used and although the section of (Benford, 1993a) describing the Rooms concept often mentions virtual worlds and space this should not be seen as necessarily implying a virtual reality interface. The paper does however give some idea of how the virtual world, and more specifically the connections between rooms, might look. Here a number of problems focused on by the criteria for this paper are addressed. The network of interconnected rooms is seen as scaling to potentially global size and supporting tools for navigation and locating people. This would be made possible by the use of tools such as the ODP Trader and OSI Directory, although the more specific nature of navigation and other tools is not investigated beyond the mention of mapping and browsing, although this should be expected as this is a single section of a paper that covers a large area. These mapping and browsing tools would help with higher level navigation while at a lower level doors would allow movement between rooms.

The design of the world therefore allows for low level connections between rooms with higher level navigation tools also available, which would be necessary for a world of such large scale. The rooms can contain not only people but tools and applications. These may be such things as shared editors or drawing tools which would be made more collaborative by the underlying communications ether of the room.

The environment proposed in the Rooms Conceptual Model is therefore one of interconnected rooms providing communications for occupants and built on an underlying architecture which encourages cooperation between CSCW applications by providing the necessary services, storage and information bases. This model satisfies many of the criteria of this document at a high level but the papers do not provide enough specific information. This however is not really a criticism of these papers as they are overviews of a very large and complex model.

1.4. Conclusions

Table 1.1 shows a very brief summary of the systems reviewed and how they fulfil the criteria defined at the beginning of the chapter. From this we can see that the majority of the systems are multi-user and make use of multiple communications media, particularly audio and video. The use of good audio links, I feel, is of great importance in synchronous CSCW systems as this provides a natural channel for passing information which can be used to resolve any problems that may occur. The use of video can greatly enhance the feeling of awareness of events at remote sites but is less important for direct communication.

We have seen a number of systems based around a rooms metaphor. This seeks to provide a natural environment that is easy to use and learn because of its similarity to the real world. One of the problems with this metaphor that does not seem to have been well addressed by the systems I have examined also comes from the real world. This is that if the group of rooms used becomes large or is not layed out clearly it may be difficult to find a way to our destination and easy to get lost. Some of the systems provide a map but these seem to be very simple and could not be used effectively for large numbers of rooms. Sadly many of the systems did not even provide this help. If a rooms metaphor is to be adopted widely much more thought must be used to overcome problems with navigation. This will apply not

System	Multi-user?	Comms.	Interface	Interaction Methods	Navigation Tools	World Building	Tailorability	Scalability
Rooms	No	N/A	2D Windows	Point & click	Map style overview. Doors. Menus.	User creates rooms and links.	Via scripts and window manager.	N/A
CRUISER	Yes	Mail, Audio and Video	2D plan with windows	Proximity. Application use.	Map. Planned and random routes.	Plan. Static after creation.	Limited. User creation of subgroups.	No distribution. Limited by navigation tools.
MILAN	Yes	Audio and Video	2D representation of 3D room. Relaxed WYSIWIS	Point & click	Individual 'village' map. Doors.	Plan. Any user may create/delete accessible rooms.	Very limited.	Distributed design. Could suffer from data inconsistency.
DIVE	Yes	Audio + VR Representation. Also some video	Immersion virtual reality	Proximity by collisions of auras.	Portals between worlds. No overview or map.	Linked individual worlds. Static after creation.	Define own body image.	Highly distributed but navigation may cause problems.
Multimedia Assistant	Yes	Audio (via telephone), mail	2D windows. Relaxed WYSIWIS	Point & click. Token Passing floor control.	Doors	Unclear who creates shared perspectives.	Some possibility in scripting language.	Designed for small numbers of rooms. Object orientation may be an advantage.
Collaborative Desktop	Yes	Audio, video (possibly), audio and mail messages	2D windows, direct manipulation	Point & click, Application use.	Team browser and catalogue.	Users may create shared room.	Define own icons and messages.	Limited to small numbers of people but has distributed architecture.

Table 1.1: Summary of the systems reviewed

only to finding 'physical' locations but also to establishing connections to other users, and so may require the use of an organisational database. The problem must be addressed not only at the level of providing navigation tools but also in the way in which the virtual world is designed, with legibility, the way an environment allows inhabitants to 'read' which route to take to a destination (Lynch, 1960), being a major concern. Our world is divided into separate domains on a number of levels: countries, cities, areas, buildings, rooms, etc. It may be that large groups of rooms should also contain separate domains to aid both management and navigation. We should however avoid too rigid definitions of hierarchies as these tend to create less desirable habitats than are provided by less rigid structures where domains may overlap (Alexander, 1989).

Another point than can be noted about the systems described is that despite their use of a very spatial metaphor only DIVE and CRUISER seem to initiate interactions between users via a spatial method and only DIVE generalises this to include interactions with tools. I see two reasons for this. The first is that although all the examples use the concept of a 'room' in some way very few have actually developed this far enough to include spatial metrics within the virtual spaces, so the users of these systems only have two states, either in a particular room or outside it. For proximity to play a part in interaction the occupants must have some attribute of position within the room. Second, most of the systems have also not carried the spatial metaphor into the interface design, preferring a two dimensional direct manipulation interface which results in interactions being initiated using the point and click method. I feel that a spatial method of interaction with the objects within the rooms would be a natural extension of the metaphor and be a fairly intuitive interface.

Many of the multi-user systems reviewed attempted to use a distributed architecture which would be an advantage if the system were to be used on a very large scale. At the same time however their designers did not seem to consider the possibility that there may be many users or many rooms in the system in other aspects of the design and so if it was really necessary to scale up the system problems would be encountered. The rooms based systems will need to consider these possibilities in all aspects of their design if they are to be used to create possibly global virtual communities. They will need to work over a variety of hardware and software platforms in a standard way but at the same time be flexible enough to allow the users to modify the applications behaviour to match their own preferences or the standard environment of their workplace.

Having looked at a cross section of systems I shall now list more explicitly some of the areas I believe have not been completely explored in work to date and so which need to be addressed by future research:

- Navigation in large scale worlds, in terms of finding both locations and other inhabitants and objects, and the possible tools that could help in this
- The related issue of the legibility of the world design and also how comfortable the users feel in the world
- Utilising the space within a room for such purposes as initiating interactions using proximity or modelling the acoustics of a room to help mediate conversations
- The concept of the boundary. Most of the systems described have used rooms as very discrete spaces with sometimes quite abstract links between them. In the real world we could also think of rooms as being defined not by the space but by the walls surrounding it. A room by this definition is part of a large space, such as a terrain, partitioned off by some boundary.
- Issues in world construction such as the static or dynamic nature of the space, who builds and owns the world and who has permission to add or delete shared rooms
- Representing people in multi-user systems to maximise the feelings of presence and proximity to other users
- The design of the underlying architecture to allow scaling up, especially in relation to emerging distributed platforms such as the ODP and OMG systems.
- The dangers of being too brutally real in the world design. This may be useful for a collection of meeting rooms but not for a system to visualise a database. In the later case we must consider how to make use of the advan-

tage of the spatial layout and familiar metaphor without compromising the goal of providing an effective representation of abstract information.

To summarise, we have seen a variety of systems built for different purposes but containing the common thread of using a rooms based metaphor. We have seen that these systems address a number of problems concerning group and individual working and employ this metaphor to overcome them. However, we can also see that the advent of higher bandwidth global networks and the increase in computing power has opened up the possibility of these systems becoming larger both in terms of size and the number of people they connect to and that this has presented new problems in the areas of navigation and world design, distributed systems, interaction between users and tools and interface techniques which need to be addressed and overcome for the systems to develop further.

This concludes our review of systems which use the rooms metaphor. Chapter two now turns its attention to the more general role of space in supporting group work. More specifically, it considers how the social organisation of space shapes the work taking place within it, using this to derive a set of requirements for the spatial model.

Appendix A: Review of Media Space Papers

A1. Introduction

This appendix will review two systems which are in may ways related to the main text and appear under the umbrella of 'Media Space'. This describes systems which provide multimedia communications facilities between a number of offices and common areas within or between organisations and buildings. These systems are similar to the rooms systems but separate because they are based strongly on real offices in real buildings rather than abstract room models. However, I believe that the criteria used to evaluate the rooms based systems will in general apply just as well to media space.

A2. Media Space at Xerox PARC and EuroPARC

The Media Space work at Xerox (Borning, 1991; Dourish, 1992; Gaver et al., 1991; Gaver, 1992; Mantei et al., 1991) is easily the most visible and probably the most important research currently happening in this area. The main paper which describes the system is (Gaver et al., 1991) while the other papers describe extensions and give evaluations of the use of media space. The main paper first describes the goals of allowing separated colleagues to work together naturally with support for both casual interaction and awareness, seamless introduction of new technology with existing systems and the understanding and support of existing work practices. The hardware is then described briefly. Most of the rooms in the research centres (the paper concentrates on EuroPARC, Cambridge, UK) contain audio-video nodes which consist of a camera, monitor, microphones and speakers. The nodes are connected together with separate data, audio and video networks.

These networked are switched by computer giving the users control via their workstations. The system is configured to provide a range of interaction methods which provide different levels of awareness and communication. These methods range from a background display of an area to full two person videophone connection. In between are a sweep of a number of offices, a glance at a particular office and office share–a long term videophone connection between offices.

The paper then goes on to voice some of the participants' worries concerning unwanted observations of their offices. The aspects of privacy that are considered are the desire for control of the connections, knowledge of being observed and the intention of the observation and the need to avoid interruption of work. Within the small community of EuroPARC a great deal of trust is used but low level control of connections is possible. A higher level service on top of this called Godard allows users to restrict access to his/her office by any of the connection methods by selecting individual names from a menu. Also, feedback in the form of auditory cues is provided when a connections to an individual's office is made.

Next the paper describes two applications called Portholes and Polyscope which allow the display of a number of video images simultaneously. Portholes is described in greater detail in (Dourish, 1992). This application was developed jointly at PARC and EuroPARC. It is designed to give users some awareness of activities in a number of physical locations by providing frame grabbed images at a slow refresh rate. The system also provides some information or properties of each of the users associated with a video view. Through these mail may be sent or other services of the media space may be accessed. The architecture of portholes is based around a number of servers which store copies of data concerning the views and the users. They are also responsible for collecting and distributing information from local devices and for collecting information from the other servers. A number of clients may access the servers, each with different services or methods for presenting the information. One of the clients mentioned in the paper simply displays a number of views selected from menus or user initialisation files while another allows for sound messages to be included as one of the properties of a view. A third client is provided for public use which can display only common areas, not individual offices. The paper includes extensive sections on the way in which portholes is used for information and awareness and the feedback and observations of the users. The polyscope system is similar to portholes but operates only with EuroPARC and has other features such as the possibility of including a few frames of animation which will periodically loop to convey some sense of movement.

After briefly summarising these systems the main paper continues by looking at a calendar based system called Khronika (Lövstrand, 1991) which notifies users of events such as upcoming meetings, visitors and video connections. This is the system used by Godard to generate auditory notification of events. Other events are also announced using sound, such as an upcoming meeting being signalled by the sound of congregating people murmuring. The main paper concludes with a summary of this and other CSCW research being carried out at EuroPARC.

Of the other papers (Dourish, 1992) has already been described. (Borning, 1991) applies particularly to this discussion as it describes work on an extension to the media space environment which is based on a virtual room metaphor. Vrooms contains a number of virtual rooms which are workstation windows containing polyscope like bitmap images but enforces symmetry as each of the people in a room can see all others who are present. Users may be present in more than one vroom if they desire and the rooms may also contain objects such as a text box (the only object implemented at the time of writing). The system is based on a client server architecture with all changes made by users being sent to the server then to other clients. Each user may create or delete vrooms or enter existing rooms. Occupants of rooms may move their images around and full video connections between two users can be established and broken by changing proximity with respect to each other. The establishment of a connection between users is signalled by their images being surrounded by a grey box. The paper continues with a review of related work and a summary of planned future work such as making communications links multi-way and the inclusion of a door mechanism for moving between vrooms. This paper also contains some details of work on the polyscope system which has been discussed above.

The final paper (Gaver, 1992) discusses the affordances of media space for collaboration. This is a critical overview of the advantages and disadvantages of video/audio communications. Some of the conclusions of this paper are that the resolution of video and the static two dimensional nature of the medium are limiting factors but media space does afford awareness and is useful as a different mode of communication rather that a substitute for face to face meetings.

Evaluation of the Xerox Media Space

The Media Space systems in use in the Xerox laboratories are designed to aid collaboration by providing convenient communications links between both fairly local and widely separated users. The ‘space’ created by the audio/video links is an attempt to electronically provide an environment similar to that of all users working in the same room, like the “electronic open plan” of the BICC Milan system. This means that occupants of the system are all provided with some sort of awareness of the activities of others, even if this extends only to knowledge of their presence or absence. The system provides a number of levels of communication with different implications concerning intrusion into the work of the person being contacted. Some methods such as the glance are designed to aid the awareness process without disturbing the subject while others such as the video phone require two way participation. Thus the establishment of a full two way link is not required for all communications activities but it was decided that some sort of notification was required and this was provided in the form of audible cues. Users may also restrict access to their offices to any subset of the people using the system. This seems to be a good compromise that provides for relatively free access but removes the risk of surreptitious observation without having to resort to enforcing full symmetry on all links.

The communications setup used by Xerox is quite impressive, with at least one video camera in each room alongside speakers and microphones. The video and audio signals are carried by separate computer switched networks to allow the use of full motion video in local areas, but this will make the system expensive and difficult to duplicate elsewhere. I would like to see a similar system established using a standard LAN. Also, the full motion video connections use a separate monitor from the computer so this requires more physical equipment in the office and separates the actual image from the controls. Longer distance links do not attempt to use full motion video and instead use very slow refresh rates. Links between users are established using buttons and, I infer, selecting destinations from a pop-up list or more directly by selecting a user from a polyscope screen. Alternatively, the vrooms system can be used to establish links by means of proximity rather than explicitly clicking on buttons or images and the rooms metaphor implies availability for interruption for all occupants of virtual rooms. All these full audio/video links however appear to be limited to two members only. Group video conferences are not possible.

The interface used in all of the Xerox media space applications are of a two dimensional click and point type. This, for the most part is quite appropriate but for the vrooms system something of a more three dimensional nature may have helped to give more of a spatial feel to the rooms as connections are made based on proximity. The interfaces here are possibly the only ones reviewed that make extensive use of sound for feedback of information from events and this appears to be more useful than, for example, flashing a message on the screen and therefore interrupting work and cluttering the screen. The ability to establish a connection with an office by simply clicking on that office's image in portholes is also a very useful interface technique as it removes the need to select the destination from a possibly long list, which must be necessary if a button is used.

The provision of navigation tools within the Xerox media space systems is not covered in any great depth in the paper. The systems seem to rely on simple selections from menu items rather than providing any automated services. For example, when using the vrooms system to enter an existing room a user must select this option from a menu and then, I assume, select the specific room from another menu (no alternative method is mentioned). When connecting to another user via, for example, the videophone the same method seems to be necessary. Polyscope provides a more elegant method by showing a wider range of possible connections but all of these methods seem to be limited to small numbers of users before they become unusable. Returning to vrooms, the system does not allow for connections to be made between rooms, although this is suggested as future work, and so movement must be made by leaving a room and connecting to another. Any user is allowed to create and delete rooms so trust is required to prevent the number of rooms getting out of hand.

The information concerning individual tailoring of the systems is sketchy. We are told that users may modify the scripts associated with buttons and so this allows for more complex operations to be built up and the functionality associated with the buttons extended. This is how the range of services from glance to videophone developed, as users tailored the basic connections to a variety of needs. Aspects of presentation are less well explained and we must assume that they are limited to what can be achieved at the window manager level.

The scaling of the system is a difficult area. This would be severely limited by the hardware and networking requirements of this media space. The use of a standard LAN for full motion video would obviously reduce this problem but is obviously nearly impossible to achieve given the bandwidth available on most networks. The basics of the system seem to rely on the computer mainly as a controller for the network connections and so where the hardware is available scaling shouldn't be too much of a problem where computational resources are concerned. At least one system, polyscope, is based around a distributed system of clients and servers where clients talk to servers which share a distributed database. There are problems however which have been mentioned earlier concerning establishing connections among large numbers of people using menu selection techniques and the polyscope system, which are caused in part by lack of screen space and also by the problems of finding an individual name in a long list. The vrooms system may solve this problem in part by the establishment of smaller subgroups but this does involve commitment by both users before the request for a connection is made and so is somewhat limited.

In summary, the Xerox papers describe a system which uses video and audio to connect a number of users in separate offices allowing a degree of background awareness of the other workers as well as full videophone facilities. Other applications build on these basics to enhance the feelings of awareness and provide more long distance links and easier access to full connection facilities. These are however limited to fairly small numbers of people due to screen space. An application based on a rooms metaphor has also been implemented and allows connections to be made based on proximity. The researchers have looked into the social aspects of the use of this system to ensure privacy and have investigates the limits of the system imposed by the technology of the system.

A3. The CAVECAT System

This paper (Manteietal., 1991) describes experiences in the use of the Computer Audio Video Enhanced Collaboration and Telepresence (CAVECAT) system and seeks to analyse this "in terms of unexpected affordances, technological obstacles and psychological impact". The paper first describes the technology. This is a very similar configuration to that of Xerox and indeed the switched network is based on that at EuroPARC and allows the same access restrictions to be applied for privacy. The only enhancement is that CAVECAT uses a 4x1 video board which allows 4 video images to be displayed on the monitor simultaneously.

The next section describes the unexpected affordances. The first of these was the use of the system by groups of people at each node rather than single users. Second was the use of the system to display ones own image as a sort of mirror. Third, people would use the system to check their own office when in another part of the building. Next the paper deals with some of the technological obstacles to the use of the system. The main problem was the delay in establishing and closing a connection to a meeting. Also the ambient noise in the offices caused problems with sound quality and the lack of directional information in the sound made it difficult to determine who was speaking. Finally, the cameras automatically adjusted to the lighting conditions but sometimes made mistakes which produced low image quality.

The paper then contains a large section on the psychological and social impact of the system. The first part of this section deal with the disparity of conversations between people with a room and conversations over the media space systems. When both were occurring there was a separation between the two types of conversation. Next, the section deals with the fact that eye contact was not really possible due to the difference in location of the monitor and camera. The third point made is the way in which CAVECAT disrupted social status in meetings due to the lack of a 'head of the table' and arbitrary placings of images within the 2x2 grid and the tendency for image positions to change if a meeting is stopped and reconvened. Related to this is the lack of cues when a user wants to speak making meeting control difficult. Next the section deals with the way the small video images influences social impact and interpersonal distance. The section finishes with a discussion of the need for access restrictions to enforce privacy when required.

The paper concludes with a discussion of the implications of these observations on the future design of the system. The main point is that many of the cues used in face to face meetings need to be duplicated within the CAVECAT interface. Also users need to be given some control over the image they see on the screen. Specific goals for the future include: metaphors to make interaction follow accepted guidelines for interpersonal communication, such as shutting a door to indicate that privacy is required; Allow more tailoring of the system by users via a visual language; more functionality such as individual control of audio and the displaying of only the speaker's image on the monitor. A final note concludes that the media space is a useful tool that not only approximates face to face meetings but provides other advantages to it's users such as awareness of events in the office.

Evaluation of the CAVECAT System

CAVECAT is, in many respects, a simplified version of the EuroPARC media space system. By simplified I mean that although the basic functionality is the same CAVECAT does not contain any of the more sophisticated interfaces used at EuroPARC, such as polyscope and the time management systems. Many of the comments that applied to the Xerox systems will, however, apply equally here so I will try to concentrate on points specific to CAVECAT.

The system described in the paper does not provide the scale of awareness that comes from polyscope and so the user specifically needs to switch between cameras to see what is happening in the office. CAVECAT does have one advantage over the Xerox system in that it naturally allows four images to be displayed on its monitor so meetings between more than two participants are easily supported up to this limit. There are problems with this method which have been covered in the paper and relate to the small size of the images. One proposed solution is to display only the image of the speaker but I feel that this will either require restrictive floor control or, if done automatically, may result in confusing switching between video images. The use of up to four way communications has however enabled the researchers to investigate the way in which media space changes the structure of meetings and this is well covered by the paper.

The interface used in CAVECAT is hardly described at all in the paper so it is difficult to comment on it. As the technology is based on that of Xerox's systems we could assume that the interface is much the same, although this is by no means confirmed in the paper. The issue of tailorability of the system is recognised as being lacking from the system and is cited as a main goal for further work and a visual language is being built to allow manipulation of the system. The paper also mentions that building media space requires addressing problems of architecture and distributed computing but does not really describe how these problems were overcome, concentrating instead on the social and psychological implications of having the media space installed.

The description of the experiences using the system show how media space removes many of the cues used in face to face meetings and this is one of the strong points of the paper and something that was perhaps lacking from the Xerox work due to its limit of two people in full videophone communication. At the time of publishing the CAVECAT work had not produced any real answers to the problem of providing substitutes to these cues but had at least defined what was lacking and what was needed for media space to provide an effective alternative to face to face meetings.

2. Requirements of Interaction in Large Virtual Spaces

Adrian Bullock and Steve Benford

University of Nottingham

John Bowers

University of Manchester

This chapter introduces key requirements which need to be considered when designing models of interaction in large virtual spaces. In so doing, it defines an agenda for future COMIC work and sets the scene for the following chapters.

The chapter begins by justifying the adoption of the spatial approach in the first place. It then explores requirements for interaction between groups inhabiting a local space. This is followed by a discussion of the requirements for constructing larger scale spatial structures to faciliate longer term and larger scale cooperative work.

2.1. Introduction

The goal of this chapter is to identify some key requirements and research issues to be considered by the COMIC work on spatial approaches to large scale cooperative environments. More specifically, it has the following two aims:

1. Justifying why a spatial approach to CSCW is a good idea in the first place. This justification is drawn from observations about the key role of space in supporting cooperation in the "real world" (some initial justification has already been supplied by chapter one which described previous projects which have followed similar approaches).
2. Considering requirements to be met by later COMIC work on spatial models of interaction.

In order to structure the work a little more clearly, the discussion of requirements in space-based CSCW systems is further divided into two main parts:

- The microscopic level — interaction within a group situated in a local space. Issues here concern how people use the properties of space to manage social interaction. Examples include managing conversations and negotiating access to objects within a given spatial setting (i.e. a single room, space or terrain). Following the primary theme of the COMIC work, we are particularly interested in 'crowded' spaces; in other words,. spaces that contain large numbers of inhabitants.

- The macroscopic level — combining individual virtual spaces into larger structures. Issues here include determining appropriate spatial structures, supporting navigation, comparing metaphorical with more abstract spatial settings, bounding space, configuring space and the persistence of spatial structures.

Where have these issues come from? Part of the input came from an informal analysis of video scenarios that were filmed at Lancaster and Nottingham in September 1992. The scenarios captured interaction within a variety of spatial settings including offices, common rooms, laboratories, major corridors and transit routes. Others have come from knowledge of previous systems. Yet others have come from the imaginations and experiences of the project team.

2.2. Why Adopt the Spatial Approach ?

Before considering requirements, we first give some justification for adopting the spatial approach in the first place. At the highest level, space has the advantage of being a natural phenomenon that is familiar to all of us — we exist in a spatial world. This familiarity allows us to fully explore the options and capabilities that space can afford to us. Space is all around us. It provides a backdrop for nearly every kind of activity we perform, as an individual or within a group. Space implies movement and travel with their associated costs and benefits. Space allows us to arrange objects, information and people. Space allows us to mark territory and ownership. Space "carries" sound and light and is intimately bound up with our abilities to hear and see. Space can be used to separate the public from the private and the personal from what is common. Space has great value and possession of space can be extremely prestigious — indeed, it is often the cause of major conflicts. All of this is because we are inherently spatial creatures inhabiting a spatial universe. As a result we are equipped (to varying degrees) with a set of powerful spatial skills that allow us to live in this world and to make some sense of it.

Exploitation of our natural spatial skills lies behind much previous work on user interface design and the introduction of metaphors such as desktops and rooms. Particular skills of interest include our cognitive abilities to spatially classify and structure information and to navigate. These skills will also play a major role in co-operative systems. However, work on CSCW is also concerned with supporting and exploiting our skills as social animals. What role does space play in our interaction with other people? What social spatial skills do we possess?

As befits work on CSCW, the following sections focus on the role of space in social interaction. In particular they explore the social significance of segmenting and bounding space; action participation and awareness in space; and the use of space in negotiating access to people (conversation management) and other resources.

2.2.1. The Social Significance of Segmenting and Bounding Space

While a full treatment of spatial phenomena in relation to social interaction is beyond the scope of this chapter, it is important for us to give a sketch of the kinds of social scientific meditations which guide our approach.

Most major social theorists have written about the organisation and social significance of space. Karl Marx wrote of how certain ways of understanding, organising and appropriating space and time emerged in the capitalist epoch (Marx, 1957). This theme is continued and extended by Lefebvre (Lefebvre, 1991) who argues, for example, that the distinction between 'private' and 'public' spaces has been differently understood at different historical moments and in different cultures (see also Massey, 1984).

In an influential treatment, which we follow in several respects, Giddens (e.g. Giddens, 1984) has discussed some of the many ways in which space and time can be regarded as a *resource* for action and interaction. He introduces the notion of 'modes of regionalization' to try to capture the interdependence of geographical space and social practices. Space undergoes different modes of regionalization, through which distinctions between different spaces (or 'locales' in Giddens' term) are produced.

> 'Regionalization' should be understood not merely as localization in space but as referring to the zoning of time-space in relation to routinized social practices. Thus a private house is a locale which is a 'station' for a large cluster of interactions in the course of the typical day. Houses in contemporary society are regionalized into floors, halls and room. But the various rooms of the house are zoned differently in time as well as space. The rooms downstairs are characteristically used most in daylight hours, where bedrooms are where individuals 'retire to' at night. (p119)

Two points about this passage are worth explicit note for they are characteristic of Giddens' treatment of space. First, he notes the different *uses* to which different spaces are put: the kitchen has different uses from the bathroom (though it must be remarked that such differentiations are quite recent in European homes and still not ubiquitous). These uses relate to the different social practices that are carried out *in* those spaces and *for which* those spaces serve as a resource: for example, tools related to cooking are kept in the kitchen, 'tools' related to cleansing the body are kept in the bathroom. The general point is that spaces are created and structured to suit a wide variety of activities. Other examples include meeting rooms, theatres, offices, sports fields, airports, waiting rooms, hospitals and countless others. In all of these, the existence of a space constitutes a key resource for establishing and enabling the activity. The same is also true for spaces which are used for transit between other spaces — corridors, foyers, squares and other spaces regarded as 'public'. Secondly and importantly, Giddens notes the inter-relations between space *and* time. In the above passage, he notes the association between regions and different times of the day. Similarly, there is a temporal rhythm to many of the other spaces we have noted. Sports fields and theatres pertain to *leisure time*. Meeting rooms and offices pertain to *work time*. And so forth. In general, Giddens and those who follow him often refer to *space-time*, rather than treat either separately.

Noting the close inter-relations between space and time, and how this connection in turns relates to social practices, is a feature of the work of 'time geographers' like Hägerstrand (Hägerstrand, 1975). Hägerstrand stresses the routinized character of much of everyday life. Everyday life has rhythms and is made up of regular, repeatable routines and practices. Importantly, these practices are spatialised and extended in time. Indeed, Hägerstrand has often emphasised that this regularity stems from a series of fundamental properties of human beings in relation to space and time which limit the forms that social organisation can take, including (the formulation of these below is borrowed from (Giddens, 1984, p.111-112)):

1 The indivisibility of the human body, and of other living and inorganic entities in the *milieux* of human existence. Corporeality imposes strict limitations upon the capabilities of movement and perception of the human agent.
2 The finitude of the life span of the human agent as a 'being towards death' [this phrase is Heidegger's]. This essential element of the human condition gives rise to certain inescapable demographic parameters of interaction across time-space. For this reason if no other, time [and hence space] is a scarce resource for the individual actor.
3 The limited capacity of human beings to participate in more than one task at once, coupled with the fact that every task has a duration.
4 The fact that movement in space is also movement in time.
5 The limited 'packing capacity' of time-space. No two human bodies can occupy the same space at the same time; physical objects have the same characteristic. Therefore any zone of time-space can be analysed in terms of the constraints over the two types of objects which can be accommodated within it.

These fundamental constraints on human activity in relation to time-space produce, according to Hägerstrand, many familiar phenomena concerning the social organisation of space, for example, the construction of routines and of 'stations' or definite time-space locations (e.g. homes, streets, cities, states etc.) where interactions between individuals routinely take place. For Giddens (Giddens, 1984), it is within the terms of and in response to these constraints that various modes of regionalization operate.

To identify the form of a region, *boundaries* are constructed (Giddens, 1984, p.121). These boundaries may be symbolic or physical. Walls between rooms, screens, customs and immigration checkpoints or marks on a map are all familiar examples of the physical-symbolic means for marking a boundary and hence constructing a region. International borders mark the territory of nations. State and regional borders mark administrative and legal territories. The elementary operation of drawing up a boundary is of utmost significance. Not only do boundaries differentiate spaces, they can serve to include some elements (people, resources, objects and so forth) while excluding others. The relation of bodies in space to boundaries within space often produces ‘front’ and ‘back’ regions. For example, at the boundary between an organisation’s building and the outside world, there may be a reception area. For many organisations, presenting a well decorated, comfortable and elegant reception area is of great importance. It is — after all — the ‘front’ of the organisation that the visitor first sees. The visitor may only get to see the ‘back regions’, if he or she is allowed in, has an appointment, is there ‘on business’ and so forth. The human body itself, of course, has clear front and back regions and these too are of interactional significance. Facing someone is a very different action from turning away. The maintenance and manipulation of front (or ‘face’) versus back distinctions in social interaction has been emphasised on numerous occasions in the work of Goffman (e.g. Goffman, 1972).

Perhaps the starkest examples of the inter-relations between bodies and boundaries come from studies of ‘total institutions’ (to use the term of (Goffman, 1961)) such as prisons and hospitals. Foucault (Foucault, 1977) provides some vivid descriptions of prisons from the late 18th/early 19th century which — he suggests — offered models for the development of most ‘modern’ state and civil institutions through principles of exclusion and division. The prison — as well as being divided off from the rest of the world — is internally differentiated in space. It is divided systematically into cells, walkways, gantries etc. For early prison designers such as the utilitarian philosopher Jeremy Bentham, these divisions were intended to have both a disciplining effect (while divided off both from each other and from the outside world, the prisoners would keep themselves in good order) and provide the opportunity for human beings to be observed in detail. Indeed, Bentham (a full account of this is given by (Foucault, 1977)) imagined that a whole science of human beings could come into existence based on observations gained under appropriate spatial conditions in specially designed prisons. The most controversial aspect of Foucault’s work lies in his suggestion that the spatial arrangements of the prison, and all that follows from this, serves as the point of reference for most modern institutions (schools, hospitals etc.) and, indeed, has insinuated itself in all aspects of modern life: we live in a ‘carcereal society’. Undoubtedly Foucault overstates this point in dramatising it and begs answers to what is a further empirical question: just how is everyday life in modern societies organised in time-space? The spatial regime of the prison is surely but one form of spatiality, not the model for all forms (cf. (Giddens, 1984, p.145-158)). Nevertheless, total institutions such as prisons provide very clear examples of how space, time and human bodies and social practices are intertwined.

2.2.2. Action in Space: Participation and Awareness

In all the examples given so far, marking out a space with boundaries is important to some social practice and ordering what bodies can and cannot do. Let us now offer the following conjecture: *boundaries in space afford different modes of participation in a social practice alongside different modes of awareness*.

Let us say a little more about what is intended here. The boundaries which separate the inside of an office from the outside enable the differentiation between who can participate as an office member and who can participate as a visitor, supervisor or whatever. The boundaries within the prison differentiate those who participate in prison life as prisoners and those who participate as warders. Etc. Etc. In addition, boundaries afford differences between modes of awareness. Again the prison provides the starkest example. The warders have an awareness of the prisoners which permits *observation* or *surveillance*. By contrast, it was Bentham's intention that the prisoners, having neither awareness of each other or of the observer, should become intensely *self-aware* and keep themselves in good order. 'Open plan' offices often employ certain forms of boundary to define separate office regions but do so in a way which enables different forms of awareness to come into existence from those possible in traditional, walled office spaces. The 'semi-permeability' of low, mobile room dividers affords a different profile of awareness of one's neighbours and their activities than does the relative impermeability of a wall.

Awareness Through Seeing 'Out of the Corner of One's Eye'

The inter-relations between space, awareness and participation are ubiquitous in working practices. Let us illustrate this with some examples taken from recent CSCW literature.

Heath and Luff (e.g. Heath and Luff, 1991) have extensively studied the activity of control room operators in the London Underground, documenting in some detail how the two operators co-ordinate their activities while maintaining their different responsibilities. This is achieved by overhearing, 'seeing out of the corner of one's eye' and other forms of monitoring of the other's conduct which establish coordination without interruption. In addition, the operators conduct their activity in such a way as to make it available to each other as potentially overhearable or monitorable. Heath and Luff have some examples of how an operator can engage in quite exaggerated body movements, gestures and ways of talking to bring his conduct to the attention of his colleague. However, in all this, it is important to note that the spatial arrangements of the control room are such that these monitorings and displays are possible. By providing the controllers with adjacent yet separate desks, they can maintain an awareness of each other's individual work without intruding upon it.

Awareness Through Seeing 'at a Glance'

The working environment of the London Underground control room operators includes a number of different artefacts — notebooks, schedules, timetables, computer screens, headsets, microphones. In addition, along the length of one of the walls of the control room is a representation of the underground line — called 'the fixed line diagram' — using a strip of lights. The location of trains is shown by illuminating the lights corresponding to the section of track where the train is. In this way, the goal of a regular, uninterrupted service can be 'seen at a glance' as a series of equally spaced illuminations. However, that it is publicly available and spatially located as it is matters greatly to the working activity of the controllers. Both can see it and both can see that the other can see it.

Arranging space so as to afford awareness through 'seeing at a glance' is a prominent feature of many working activities. In studies of Air Traffic Control, Hughes et al. (Hughes, Randall and Shapiro, 1992) show how 'flight strips' spatially arrange information about air traffic, how these strips themselves can be arranged in a rack to signal problematic flights, how flight control suites themselves lay out the strips and other resources so that the practised on-looker can see how smoothly or otherwise the work is going. Similar points about the inter-relations of spatial arrangements, visibility, awareness and intelligibility have been made by Anderson and Sharrock (Anderson and Sharrock, 1993) in a study of the processing of invoices in an entrepreneurial firm.

While these are just two sets of examples from emerging research in CSCW, they hint at a general point. Space and spatial arrangements of bodies, devices and various forms of representation (charts, diagrams and so forth) are important for sustaining various forms of interaction, participation and awareness.

2.2.3. Negotiation in space: conversation management and sharing resources

The implicit awareness of the presence and activity of others afforded by space provides the basis of a whole range of subtle negotiation skills among its inhabitants. Put another way, humans have developed a range of spatial social skills to complement their cognitive skills. The essence of these skills is that awareness of others allows people to modify their own behaviour in social situations. It also provides a degree of predictability concerning the likely actions of others (you can easily see when someone is heading across the room to talk to you or when they are heading for the door). The result is a kind of social dance where people subtly employ movement, position, orientation, gesture, body orientation and other devices in order to negotiate interaction. Key features of this negotiation are the high degree of autonomy, yet inter-dependence, enjoyed by participants, the extremely rapid feedback provided and the fact that social mechanisms of control gradually emerge, evolve, adapt and can be broken when desired. We now consider two specific cases in more detail: negotiating "access" to other people and negotiating access to non-human resources.

Negotiating access to people — conversation management

Conversation management is a prime example of negotiating access to people. This is a subtle art which covers joining and leaving conversations and negotiating turn-taking and repairs. Space assists in joining conversations by signalling the gradual approach of other people. This allows both the approaching and the approached groups to gradually adjust their behaviour to accommodate the others (e.g. changing tack away from a sensitive subject and making space for the new participants). Space also allows us to signal that we don't wish to be interrupted either crudely through devices such as closed doors or more subtly through gesture and positioning. We also use space to signal that we wish to interrupt someone (e.g. hovering near their shoulder). Within a conversation we use properties of space to show who is speaking and who wishes to speak next. In particular, orientation, gesture and gaze direction can be used as flexible mechanisms for the control of turn-taking. Finally, when we are talking, we can judge the reactions of others from their use of space (e.g. restless shifting around signifies boredom and suggests that we should give way to someone else).

Negotiating access to resources

Space also supports us in negotiating access to shared resources by providing awareness of who is using them and who intends to use them. Consider a whiteboard. As a communication tool it has no built in control mechanism to stop conflicting use (e.g. two people drawing over the top of each other's work). However, its placement in a space means that we are instantly aware of who is using it and also who is approaching it. This enables us to develop social conventions for controlling access to the resource. The world provides us with many examples of such mechanisms which result in behaviours such as queuing, jostling and scrumming (the exact behaviour tends to vary with culture).

2.2.4. Space and the unexpected

As a final observation, a key characteristic of some spaces is that they allow the unexpected to happen. This is particularly true of public spaces where you might meet anyone at any time. This notion of chance encounters or of bumping into other people may play a key role in initiating interaction and in promoting social interaction. Indeed, this approach underpinned the development of the CRUISER system and also lies behind a number of media-space developments (e.g. the Glance mechanism).

To summarise so far, human beings are fundamentally spatial organisims and space (indeed, space-time) is intimately bound up with everything we do. As a result, we have developed a rich set of spatial skills, both cognitive (e.g. spatial classification and navigation) and also social. Our discussion has focused on the latter, reflecting on the many ways in which space impacts on our interaction with each other and our activity in social practices. We have considered the social significance of bounding and segmenting space; action, participation and awareness in space; and finally people's use of space in negotiating access to each other and to common resources. These observations provide a compelling motivation for adopting a spatial approach to CSCW. In essence, our goal is to provide a powerful set of mechanisms which afford the same kind of flexibility as space allows in cooperative work.

The following section begins this process by abstracting key features from the previous arguments into a set of requirements.

2.3. Requirements and Issues

This section presents a list of requirements to be considered when designing spatial models of interaction for CSCW. Collectively, these requirements define a research agenda for later modelling work.

2.3.1. The microscopic level — cooperation within a space

We start by considering requirements which apply to interaction within local groups situated in space. We refer to this as the "microscopic level" as it is concerned with the details of interaction between individual group members.

Awareness

First, we need to provide mechanisms for propagating awareness across space. Awareness should include informing a person who else is present, where they are located, what they are doing and what their capabilities are. At one extreme, awareness might be specific and focused. At the other it might be peripheral. Many different kinds of thing can be an object of awareness. One can be aware of people, resources, objects and events in space. Awareness might be propagated through many media. Flexible mechanisms should be provided to allow people to control both their awareness of others and others' awareness of themselves. A range of awareness states should be enabled such as overhearing, monitoring, observing, ignoring and others.

Embodiment

People should be sufficiently and appropriately embodied in space. This means that they should be provided with body images which convey presence, location and activity to others. People may have body images in many media, not just visual bodies. Body images should convey personal identity and should be tailorable. At the same time, they should be truthful and should not suggest false identities or the presence of abilities that the person doesn't possess (e.g. don't give them "ears" if they can't process audio input). There is clearly a tension between these last two requirements. Body images need not be highly realistic and human in appearance — they might be quite abstract. However, body images should have identifiable 'front' and 'back' regions, so that bodily orientation can be used in the coordination of interaction. As an additional observation, many of these requirements also apply to the images of non-human objects.

Gesture and body language

Related to the issue of body image is that of gesture and body language. People should be able to develop and use such spatial signals in order to enhance interaction. Given that body images may not be human in appearance, we don't preclude the development of entirely new forms of gesture.

Movement and transit

People should be able to move at will through space. Furthermore, this movement should involve a sufficiently high number of degrees of freedom (e.g. up to six degrees in 3-D space — 3 translational and 3 rotational). On the other hand, it is not clear whether people should have to experience movement in order to reach their destination. Unlike Michael Benedikt who proposes a "Principle of Transit" for cyberspace where all movement occurs phenomenally through all intervening points in space with some cost which is proportional to the distance travelled (Benedikt, 1991), a principle which bears comparison with Hägerstrand's observations of time-space correlations, we wish to entertain the notion of teleportation in virtual space. We regard it as an open question as to whether we need reproduce *all* the spatial constraints of the real world in the virtual one. However, we also recognise that transit offers several advantages such as exploration, learning about new environments and the chance to bump into people. We also recognise that teleportation may result in some interesting social problems such as jumping into the middle of a group of people who are talking about you. Some other mechanisms may be required to solve these problems.

Views, perspectives and spatial frames of reference

People's views should be sufficiently synchronised so as to facilitate cooperation without resulting in complete loss of autonomy. This idea of a view encompasses scale, orientation of objects, the presence of objects and the locations of objects and people within the shared space. In particular, Koike (1992) describes how multiple views onto the same object within three dimensional space can highlight relations between objects and users that would otherwise be missed. On the one hand, collaboration requires some overlapping of view (e.g. in order to use spatial terms in conversation). On the other, we wish to avoid the rigid constraints of the WYSIWIS paradigm and the resulting need for cumbersome floor control mechanisms to drive shared user interfaces.

A key requirement is to separate having a common spatial frame of reference from adopting individual perspectives within this frame (i.e. individual positions and orientations). Collaboration may require a common spatial frame. It may actually be hindered if perspectives are also synchronised. The problem with WYSIWIS is that it fails to make this distinction.

Conversation management

People should be provided with flexible and natural spatial mechanisms to manage conversation. These mechanisms should support the management of turn taking within a group and joining and leaving conversations.

Negotiating access to Shared Objects

People should be provided with flexible and natural mechanisms for negotiating access to shared objects in space. Furthermore, these mechanisms should support the evolution of social control mechanisms such as queuing, jostling, hovering and

scrumming. They should also indicate the likely availability of the object in much the same way as if we see a large queue for a “real world” object or person.
We finish this section with a few more general requirements relating to those introduced so far.

Scaleability

Spatial models of interaction should be scaleable. This means that they should naturally scale to support interaction among large groups in virtual space.

Autonomy and Inter-Dependence

Spatial mechanisms of interaction should preserve autonomy as much as possible. This is particularly true for conversation management and turn taking where individuals should not be prevented from “speaking” through crude technical measures, even if no one is hearing them. Floor control mechanisms which remove this right are tantamount to gagging people and then letting them speak by removing the gags for short intervals. This seems to be a crude technical mechanism for managing what in everyday life is accomplished by means of a natural social ‘mechanism’ — the turn-taking machinery of ordinary conversation. There seems to be no *a priori* necessity to provide aggressive technical solutions when we can expect routine forms of social practice to emerge.

While it is true, then, that ordinary social interaction provides a certain autonomy to participants, this autonomy is intimately connected to the inter-dependence that participants have one with another. The significance of a turn in ordinary conversation, while that turn may be in a sense ‘freely’ given to the conversation, will be heard in the light of other turns that have been issued in the conversation and in the light of the expected, ‘normal’ operation of the turn-taking machinery of conversation. Thus, while individual contributors are autonomous in the sense that they are *in principle* free to contribute as and when and what they like (there are no gags), *in practice* they are ‘caught in a web’ of inter-dependencies, norms and expectations which they have to attend to or which others may hold them to account for if they don’t. It is important to investigate whether it is possible to respect this subtle mixture of autonomy and inter-dependence in CSCW systems based on action and interaction in virtual spaces.

Power balance

Spatial mechanisms of interaction should maintain a power balance between “speakers” and “listeners” in any given interaction. More specifically, speakers should be able to influence the scope of projection of an utterance and listeners should be able to influence its scope of reception. The implication of this requirement is that control of whether a given utterance is heard should reside with both the speaker and listeners. Or in other words, interaction should be treated as truly a social phenomenon and not a mere series of isolated, individual actions tied end to end.

Mechanisms vs. rules

Spatial *mechanisms* of interaction should be differentiated from *rules*.. Although the distinction is not always clear, a mechanism is generally intended to be free of semantic meaning whereas a rule implements a particular communication policy. By way of example, a rule might prescribe a specific sequence of turns in a conversation or might assign a particular type to an utterance, whereas a mechanism provides a relatively neutral basis for developing social conventions. In essence, rules implement social conventions within a system whereas mechanisms are intended to support their development and evolution external to the system.

The principle of extended symmetry

The principle of extended symmetry proposes that mechanisms that apply to humans should also apply to other objects inhabiting virtual space. That is, humans and non-humans are to be treated symmetrically (at least in principle) in a virtual world. Thus, all objects might be aware of others, might compete for resources and might negotiate access to each other. While this principle might seem counter-intuitive, it is important as a design requirement. As remarked above, it is an open question whether a virtual world must replicate *all* the constraints of the real world. Allowing more flexibility in the design of virtual worlds seems important to us, as more tightly constrained worlds can be regarded as particular ‘parameterizations’ of a general approach. Treating humans and non-humans symmetrically in a virtual space allows for possibilities such as ‘active objects’ which are ‘aware’ of events and so forth in a virtual world. It does not seem justified to us to rule these possibilities out and expressing the principle of extended symmetry as a requirement is the clearest way to remind ourselves of these possibilities.

2.3.2. The macroscopic level — structured spaces

The previous section discussed cooperation within a single space, focusing on how the properties of space can be utilised to manage conversations, enhance collaboration and provide a natural setting for interaction. We now go on to consider the requirements of cooperation in larger more structured spaces. We call this the macroscopic level.

Dimensionality

It is important that we provide a suitably general definition of space so that our concepts and models can be applied across a wide range of systems. First, any given space may be defined in any number of dimensions. Clearly, we are used to thinking of two and three dimensional spaces, however, we should recognise that other dimensionalities, particularly order higher ones for information spaces, are possible. Second, these dimensions may be graduated in either discrete or continuous ways.

Real world metaphors vs. abstract structures

The structure and layout of a virtual space will impact on the nature of the work taking place within it (in the same way that physical architecture affects our interaction within real-world rooms and buildings). At a more fundamental level, we identify the requirement for two general approaches to building such spaces: the use of real-world metaphors and the definition of abstract spatial structures.

The metaphorical route is the most immediately obvious approach. Why not borrow familiar concepts such as rooms, buildings, doors, windows, streets, districts and cities? The potential advantage of this approach is great familiarity and easy learnability for users. Of course, the risk is that of reproducing unnecessary real-world constraints in the virtual world (e.g. having to move through corridors to get to rooms, or buildings whose design caters for a non-existent gravity). Chapter four discusses the advantages and disadvantages of employing real-world metaphors in the design of CSCW systems in more detail.

The abstract approach centres around the notion of abstract data spaces, called "information terrains" which allow users to inhabit the data which they are using and discussing. We can identify a range of techniques for constructing these terrains from areas such as scientific visualisation, database visualisation and information science (see chapter three).

Boundaries

There is requirement to be able to define boundaries in space in order to mark spaces for activities, mark territory and facilitate privacy and security. Boundaries definition should be a flexible and dynamic process. Boundaries will affect spatially related phenomena such as awareness and transit. More specifically, they may be permeable, impermeable, semi-permeable and even conditionally permeable to awareness and transit. These effects may vary with different media (e.g. a "window" boundary may be permeable to visual awareness, semi-permeable to audio and impermeable to transit). Boundaries may be fixed or mobile. Indeed, highly mobile objects such as people may act as boundaries (consider a row of police with interlocked arms controlling a crowd).

Personalisation and ownership

Ownership and possession of space can be of great importance to human beings. Some spaces (but not all of course) are 'owned' and the ownership of space can exert an important influence over the forms of action and interaction that can take place. As a result, there is a need to be able to mark space so as to reflect ownership. One approach is to draw up boundaries of a special sort. Another, reflecting a weaker form of 'ownership', is to allow people to personalise the spaces they inhabit (examples include the placement of notices, posters and other personal effects and even graffiti).

Navigation

It is a mistake to think that people always know where they want to go or how they can get there. Navigation will be a key requirement in large spaces. Navigation covers both exploring environments to see what is there and finding the way to a particular destination. One can envisage many techniques for supporting navigation including:

maps (both directional and general) — a representation of the world which allows the user to visualise relationships between different spaces and so navigate around. Maps may reflect both spatial structure and dynamic activity.

guides and tours — a pre-programmed guide can be used to show users around the space they are occupying.

signposts and labels — these can show distances and directions to major landmarks within the space as well as providing information on them. Different levels of information can be given dependent upon proximity to the object pointed to by the signpost (e.g. road signs on a long distance journey) (Ingram, 1993). Indeed, maps are generally useless without signposts.

histories — a record is kept of where you have been and by using this you can find your way back. Examples from history include the myth of Theseus and the Minotaur where Theseus used a piece of string to trace his route so he could find his way out of the labyrinth easily.

Legibility

The disciplines of urban planning and architecture teach us that the structure of an environment can do much to help or hinder the process of navigation. In his book "The Image of the City", Lynch introduces the term "legibility" to refer to the inherent graspability of a given environment (Lynch, 1960). He also proposes several techniques for enhancing legibility including the use of landmarks and accentuating nodes, key pathways and the distinct character of different districts. Legibility of virtual environments is a key requirement and we may be able to borrow and apply many of Lynch's concepts to both metaphorical and abstract spaces.

Configurability

The inhabitants of worlds should be able to change them to reflect their needs. This might include creating new spaces, manipulating boundaries, sizing and shaping space and adding new objects into space.

Persistence

Spaces should persist over extended time periods in order to be believable. In particular, it should be possible to configure and redesign spaces from the inside without having to take them off-line.

2.4. Summary

This chapter has laid the foundations for later work by describing the motivation for adopting a spatial approach to CSCW and key requirements to be met by the resulting spatial models and systems. The motivation stems from the observation that humans possess powerful spatial skills, both cognitive and social, which might be exploited in the design of CSCW systems. Particular attention was paid to social skills in the area of conversation management and negotiating access to shared resources as well as to the social significance of bounding and segmenting space. The requirements effectively define a research agenda for later work and have been divided into two main levels.

The microscopic level — managing interaction in local groups

Key requirements include support for:

- promoting awareness;
- appropriate embodiment of people;
- gesture and body language;
- movement and transit;
- individual perspectives within common spatial frames of reference;
- conversation management; and
- negotiating access to shared resources.

Furthermore, these should be considered within the context of the broader requirements of scaleability, preserving autonomy and inter-dependence, maintaining a power balance between speaker and listeners, and encouraging symmetry between the treatment of people and other objects.

The macroscopic level — inhabiting larger spatial environments

Key requirements here include support for:

- constructing spaces of varying dimensions;
- building spaces based on real-world metaphors and also more abstract data spaces;
- defining boundaries in space;
- personalisation of space so as to reflect territory and ownership;
- navigation;
- improving the legibility of spaces; and
- enabling the long term persistence of spatial structures.

Chapter three now turns its attention to developing an initial model of group interaction within virtual spaces which aims to meet some of these requirements.

3. Viewpoints, Actionpoints and Spatial Frames for Collaborative User Interfaces

Steve Benford

University of Nottingham

Lennart E. Fahlén

SICS

This chapter analysis previous "shared space" systems from the point of view of synchronization of user interfaces. In turn, this analysis leads to a new design framework for collaborative user interfaces which can be used as the basis for later modeling work.

Synchronization is a key issue for collaborative user interfaces. An examination of current approaches, in particular the concept WYSIWIS and the use of Video as a communication medium, highlights a number of problems in this area including lack of a common spatial frame of reference, lack of appropriate embodiment of users and inflexible and rigid communication channels between users. The chapter proposes a new framework for designing collaborative user interfaces which addresses these issues. This framework is based on the notion of a common spatial frame within which embodied users are free to move autonomously, being casually aware of each other's activities. Embodiment is considered in terms of both individual viewpoints and actionpoints (e.g. telepointers) within the display space. The framework proposes that, in many cases, synchronization of the spatial frame is necessary but synchronization of viewpoints and actionpoints may actually inhibit collaboration. The chapter finishes with the argument that adoption of this model requires a long term transition from two dimensional to three dimensional interfaces and also the adoption of "softly enforced" social conventions in place of rigid interaction protocols in order to support natural and effective collaboration.

3.1. Introduction

Previous chapters have considered the use of the rooms metaphor in CSCW and related systems and also how the social organisation of space reflects and shapes real-world cooperative work. This chapter completes the background to the COMIC spatial modeling work by considering the issue of synchronisation in "shared space" CSCW systems. More specifically, it focuses on the limitations of the WYSIWIS paradigm in shared-screen systems and of video as a communication medium. The result of this analysis is the proposal of a new design framework for synchronisation in shared virtual spaces which can be built upon by later modeling work.

Collaborative user interfaces, particularly shared workspaces, have been the focus of considerable research effort in recent years. Resulting systems include multi-user editors and drawing tools (Ellis, 1991; Foster, 1986; Greenberg, 1991), shared screen systems and more specialized design surfaces (Ishii, 1992). There has also been a growth in the use of multi-media technology to support communication and awareness between the users of such systems including conferencing systems (Sarin, 1985) and media-spaces (Gaver et al, 1992; Root, 1988). Determining an appropriate degree of synchronization across multiple user interfaces is a key issue for such systems. At one extreme no synchronization, where users may hold entirely unrelated views of the universe, may make collaboration virtually impossible. Perhaps more surprisingly, the other extreme, that of complete synchronization, may also inhibit collaboration due to loss of individual autonomy and the overhead required to maintain common views. The nub of the synchronization issue rests on the tension between group and individual needs, an issue across the whole of CSCW.

This chapter proposes a theoretical design framework for collaborative user interfaces that aims to achieve an appropriate degree of synchronization. To peer ahead for a moment, this framework will stress the importance of providing a common *spatial frame* of reference for multiple users and the separation of this frame from the selection and presentation of displayed objects and the individual perspectives of the users. Our use of "space" in this context is in a very general sense, i.e. an arbitrary mapping of any kind of dimensions or axis of information into a landscape where some useful sense of location, direction and movement can be established (Benedikt, 1991). The framework will also focus on the issue of appropriate embodiment of users, including the separation of their *viewpoints* from their *actionpoints*. This will be combined with promotion of awareness of the presence, identity and activity of others in a common space, encouraging the replacement of communication rules with "softy enforced" social conventions. Put in other words we suggest the use of an "open" design methodology, where the systems and/or applications resources are summoned to enforce the users communicative capabilities. Restrictions and rules are imposed only if really necessary and then in away that is as flexible and under user control as is possible. However, we first begin by considering some critical limitations of two current approaches to collaborative user interfaces, the WYSIWIS paradigm and the use of video as a communication medium.

3.2. Problems with WYSIWIS

WYSIWIS ("What You See is What I See") prescribes a tightly coupled way of synchronizing multi-user interfaces and has been proposed as a foundational concept for shared systems (Stefik, 1987a). Under WYSIWIS, users see the same objects, in the same ways and in the same places. WYSIWIS is also often enhanced by the addition of individual telepointers.

We argue that WYSIWIS may in fact inhibit collaboration by binding people too closely together. In effect, it places too many constraints on people's freedom to move and act as autonomous individuals. Furthermore, maintaining synchronization during concurrent multi-party activity requires considerable effort. This effort is either passed onto users or onto the underlying system, the latter in the form of complex locking mechanisms and heavyweight and costly networking protocols (which may in turn inhibit the establishment of shared workspaces over wide area networks). For example, turn-taking mechanisms control synchronization in systems where there is one "writer" and "many" readers (e.g. (Greenberg, 1991)) and more collaboration aware systems often introduce explicit mechanisms for managing the sharing of displays (Rein, 1991).

Similar reservations about WYSIWIS are evident in the work of other researchers, leading to the notion of "Relaxed WYSIWIS". In particular, Stefik *et. al.* have proposed four dimensions along which WYSIWIS might be relaxed (Stefik, 1987b). These are *space*, where WYSIWIS is only applied to only a subset of visible objects; *time*, where delays in updating views are allowed; *population*, where sharing may be limited to subgroups of the user population and *congruence*, where alternative views (e.g. visual variations) of objects may be possible. These steps go some way towards providing users with a degree of autonomy while maintaining a degree of synchronization. However, we believe that relaxed WYSIWIS still imposes a fundamental and unnecessary constraint on users in that it synchronizes their viewpoints within the display space. Put another way, it effectively glues their heads together. This falls into the trap of assuming that looking at a common object implies looking at it from the same perspective.

This approach can be contrasted with that of Shu and Flowers who, in their work on collaborative three dimensional design, argue that users should have autonomous but mutually visible *viewpoints* within a common space (Shu, 1992). A viewpoint represents a users position within the display space and also their orientation and hence their focus of attention or gaze direction. Thus, viewpoints combine both the position that the user is looking at and the position that they are looking from within the display space. A further key aspect of viewpoints is that users are aware of each others viewpoints in a shared space and thus have some notion of where other people are working. Put more formally, viewpoints directly *embodyi* users within the display space (i.e. give them a visible presence).

We can contrast the notion of *viewpoint* with the use of telepointers in shared workspaces. Many WYSIWIS systems support the notion of autonomous telepointers which allow different users to indicate places of interest within the common display space. Like Shu and Flowers' viewpoints, telepointers aim to embody users within the space, to show where they are working and what they are currently doing. The difference between viewpoints and telepointers is that viewpoints represent where users are looking whereas telepointers show where they are manipulating.

Drawing on our experience of interaction in the real world, viewpoints correspond to the position of our head and eyes where as telepointers correspond to our hands. In fact, telepointers are just one example of what we chose to call *actionpoints* (see below). Furthermore, we note that viewpoints and actionpoints are not necessarily coupled together in a single individual and that they are rarely synchronized between different individuals. However, other people are generally *aware* of where we are looking and manipulating. In other words, real world interaction supports both autonomous viewpoints and actionpoints coupled with awareness of the viewpoints and actionpoints of others.

3.3. Problems with Video

Now we turn our attention to the use of video in collaborative systems, particularly its use within teleconferencing and media-spaces. Although the use of video may seem to be a separate issue from that of WYSIWIS, we argue that video suffers from similar fundamental limitations. The introduction of video between remote participants may be intended to meet several goals. First, video may enhance audio in teleconferencing by allowing users to see each other, to make eye contact and to more effectively manage conversation (e.g. effecting turn taking and repairs) (Acker, 1987). Second, video may be intended to enhance general awareness through the introduction of facilities such as glancing and office-sharing in media-spaces (Gaver et al, 1992; Root, 1988). Gaver has already discussed the affordances and some of the current limitations of video in some depth (Gaver, 1992). This chapter focuses on the following key limitations:

1. Lack of a common spatial frame — although video may link together separate physical spaces, it fails to provide a consistent common spatial frame between them. In other words, there is no common navigable space established in a video connection. Instead, video provides a non-traversable window (and usually a small one at that) between separate spaces. Participants using video therefore cannot easily establish common spatial terms of reference. As a more concrete example, the notion of making eye-contact in a video conference of more than two people is plainly nonsensical. Glancing at a camera will result in the glance being transmitted to many people — just who is being glanced at? Contrast this with the ability to tell who is attending to who around a real conference table by observing gaze direction. Interestingly enough, this problem does not become apparent until we consider groups of more than two people.

2. Lack of embodiment — people using video act as external observers peering into a space through a window. They are not actually embodied within the remote space. This may result in several problems. First, they are unable to easily move about in the remote space, experiencing it from all angles and perspectives. Second, lack of embodiment may cause social problems in the use of video technology (e.g. people feel that the are being watched and are unaware of the observer's presence or identity). Perhaps the general case is not so much one of sharing viewpoints, but is more how you perceive and relate to other collaborators. It seems that it is important to have a casual and non-threatening trustful awareness of others. The video

tunnel's "en face" view is not sufficient and might even be considered harmful (Rodden, 1993).

3. Lack of model — a digital video signal is essentially a stream of bits, with no higher level model of what the bits represent. Consequently, manipulation and reconstruction of video images is extremely difficult. Although advances in feature recognition and scene analysis may represent a small step in this direction, video still suffers from an inherent lack of flexibility. For example, one cannot easily move or change the appearance of objects in a video image. As an analogy, one could liken the use of video in CSCW systems to building a drawing package by using only scanned photographs of shapes. The use of video also suffers from several further shortcomings such as lack of continuity, modularity, scaleability, and extendibility together with an over reliance on technical artifacts. When these are considered in combination with the lack of consistent spatial frame, the authors feel that the use of video should be treated as an extension and enhancement to the conventional telephone and not as the *basis* for building general purpose computer based wide area CSCW environments.

3.4. A Framework for Collaborative User Interfaces

This section proposes a design framework to address the issues raised above. The aim of this framework is to provide a clear separation between key design issues and to show how they relate to each other, thereby allowing designers to determine appropriate synchronization policies. To summarize, previous sections have identified seven key design issues for collaborative interfaces:

1. Both too little and too much synchronization of users displays may inhibit the ability to co-operate. In particular, WYSIWIS involves too much synchronization.
2. People should possess independent viewpoints and actionpoints within a common display space.
3. Viewpoints and actionpoints need not be synchronized between individuals. However, people should be aware of other peoples viewpoints and actionpoints.
4. People should be aware of each others presence, identity and activity (the latter supported via viewpoints and actionpoints).
5. A common spatial frame should be provided. Existing multi-user interfaces typically lack a common spatial frame. Instead, they provide synchronization between sets of disjoint spaces.
6. Users should have an embodiment. Many existing interfaces fail to adequately embody users within the common display space.
7. Firm and fast mechanisms (or rules) for management and control of shared workspaces should be replaced by social conventions that are not *enforced* by the environment itself.

We propose a new design framework for collaborative interfaces which addresses these issues. The framework consists of the following components.

3.4.1. Components of the Framework

Selection

A collaborative system may contain many objects, of which only a subset may be visible at a users interface at a given time. Selection describes the process of choosing which objects are represented from the set of possible objects. A requirement on selection is that it should be lightweight and non-obtrusive.

Presentation

Once selected, it is necessary to choose the mode of presentation of an object. In particular, each visible object may have many possible representations (e.g. a clock representation might be digital or analogue or a representation of temperature might be either color or length/height). It should also be possible to agree on common representations. An architecture for managing multiple presentations of objects has already been proposed in (Bentley, 1992).

Spatial frame

The spatial frame refers to the spatial frame of reference used in each person's display mechanism. The spatial frame locates all objects within the display space by defining a co-ordinate system which allows the measurement of both position and direction. The spatial frame may be defined to have any number of dimensions.

The notion of a *common* spatial frame means that users inhabit the same co-ordinate system. Consequently, they can assign consistent relative positions to each other and to all other observed objects. In turn, this supports the use of consistent spatial language and conventions (e.g. if I observe that person A is looking at B, I can be sure that this is also true for you, for A and for B).

Embodiment

Embodiment describes the way in which users are themselves directly represented within the display space. Notice that we consider users as existing *within* the space, not as observers looking onto it. Note that this does not necessarily imply the use of immersive virtual reality technologies, only that the user has some representation within the space. Embodiment conveys awareness of presence, identity and even activity to other people and is therefore fundamental to managing cooperative work. As a result, embodiment must satisfy a number of possibly conflicting goals including personalisation (i.e. easily conveying identity and allowing people to tailor their own images), the identification of position, direction, activity and also what can be called truthfulness (it may be confusing and harmful for objects to appear other than they are — e.g. lying about identity or suggesting capabilities not actually possessed, for instance showing ears when an object cannot hear).

Viewpoints

A viewpoint represents where a user is attending within the display space (i.e. an individual's perspective). Thus, a viewpoint is analogous to the position and direction of a person's gaze within the real world.

Actionpoints

An actionpoint represents a point within the display space where a user is interacting with some object. Examples of actionpoints are telepointers on a shared drawing surface, cursors on a shared editing surface and "hands" within a virtual reality system. Thus, we separate the point of attention (viewpoint) from the point of manipulation (actionpoint). This is analogous to the separation between our eyes and hands within the real-world. Following on from this analogy, viewpoints and actionpoints may be independent of each other, although they may often be used in a loosely coupled way.

Awareness and communication

Awareness of the presence, identity and activity of others in a shared space is critical for establishing and subsequently managing cooperative work. Awareness may be peripheral or focused. At the extreme, awareness leads to direct communication via some appropriate medium. Awareness relies of embodiment and includes awareness of both the viewpoints and actionpoints of others.

3.4.2. Synchronization of components

Having identified key components of collaborative interfaces, we now return to the issue of synchronization. More specifically, we consider the question "which of these components should be synchronized in order to best support cooperative work?" Although we argue that there is no hard and fast answer to this question and that the degree of synchronization will depend upon the preferences of the people involved and the nature of the application, we propose a core solution which we believe will act as a guideline or starting point for most cases. As stated above, we believe that some synchronization is needed, but that even relaxed WYSIWIS is too extreme. We are now in a position to phrase this argument more formally.

> *The key problem with WYSIWIS is that it does not allow autonomy of viewpoint between different users within the common spatial frame (although it does usually allow autonomy of actionpoints).*

As for video, we can state the main problems as follows.

> *The key problem with video is that it provides no common spatial frame of reference within which users can be embodied and actionpoints and viewpoints can be provided*

As a result we propose the following general synchronization policy:

1. Spatial frames should be synchronized. This means that multiple-users should experience a common spatial frame of reference, allowing them to consistently place each other within the shared space and also to use spatial terms during inter-

action. It is important for cooperative work to establish at least one common spatial frame between participants.

2. Individual viewpoints and actionpoints should not generally be synchronized between different people. In other words, each person should be free to control their own view points and actionpoints. There may be some cases where synchronization would be useful (e.g. riding piggy-back and following someone else's view of a complex task).

3. Selection and presentation can be synchronized to varying degrees depending on the application. Thus, the extent to which users see the same objects presented in the same ways may vary. In particular, the framework supports the idea of multiple presentations of shared information as discussed in (Bentley, 1992). In general, the less synchronization of selection and the less synchronization of presentation between users, the harder it will be to collaborate.

It is also important to consider how synchronization is applied. Instead of enforcing synchronization through hardwired mechanisms, support for awareness and communication allows participants to negotiate synchronization with each other. Such negotiation may either be direct or may be indirect, the latter through a process of observing the position of others in the common spatial frame. Thus, firm rules and heavyweight mechanisms are replaced by "softly enforced" social conventions which are "applied" by users not by the computer. This equates much better to the way in which groups manage access to resources in the real world. For example, a whiteboard has no internal mechanism to limit or control multiple access and turn-taking. Instead, people subtly negotiate access based on an awareness of who is using and wants to use the whiteboard backed up with social convention (it is generally rude to interrupt people). In turn, this mechanism is supported by continual awareness of who is attending to and manipulating the board within a common spatial frame. The social conventions can be broken, although at some social risk.

3.4.3. Sub-spaces and unfolding

Many user interfaces support the notion of subspaces (e.g. nested windows). We can relate this to our model through the concept of "unfolding". In his work on virtual reality, Michael Benedikt introduces the term unfolding to refer to the creation of sub-spaces within a Cyberspace, where each subspace is defined by its own set of dimensions (Benedickt, 1991). Our framework should also allow the creation of sub-spaces through a process of unfolding, where each subspace defines its own common spatial frame and may provide its own synchronization policy. The idea of having different synchronization policies in different (sub) spatial frames provides our framework with further flexibility.

3.4.4. Reflection

Let us briefly reflect this model back onto a variety of different applications. First, we consider a virtual desktop application where the display constitutes a moving window passing over a large 2-D surface containing windows, icons and other objects. The virtual desktop itself represents the common spatial frame and should thus be synchronized across multiple users. However, the position of a given display and pointer/cursor should be autonomously managed. Furthermore, the positions of other peoples viewpoints and actionpoints should be made visible in some form so as to provide some sense of embodiment. A major difficulty here might be the limited view afforded by the display which fails to provide peripheral awareness of the presence and location of others. This might be solved by the provision of a separate map facility which affords an instant overview of the entire virtual desktop surface.

A second application area might be that of video-conferencing or media spaces. Here the major problem to be addressed is the lack of a common spatial frame. Some mechanism is required for situating a set of video images within a common space and for navigating this space. One possible approach is suggested by Bill Gaver (e.g. the use of Mobile Cameras) (Gaver, 1992). Another might be to situate the two dimensional video images within some kind of 3-D display space. This leads us to a third application domain, the area of multi-user 3D based virtual reality systems, as exemplified by the DIVE (Distributed Interactive Virtual Environment), system developed at SICS (Carlsson, 1993b). A participant in a DIVE world is either a human user or an application process. Users can navigate in 3D space and may see, meet and collaborate with other users and applications in the environment. The user is represented by a body-icon, to facilitate the recognition and awareness of ongoing activities. This body-icon is an user supplied graphical object of arbitrary complexity. In most cases this body-icon, for the reasons outlined is this chapter, at least has a discernible back, front, up and down, together with identifiable manipulative actuators, i.e. it is easy to see the “face” and the “hands” and associated view- and actionpoints. Furthermore, in DIVE a body-icon (or any object for that matter) may have some autonomous statemachine driven behaviors associated with it to further aid the animated graphical modeling of a user in 3D space. The concept of unfolding is handled by passing through “gateways” between logically different spaces or “worlds”. A sophisticated use of the DIVE system according to the presented framework is outlined in (Fahlén, 1993), (Carlsson, 1993a) and (Benford, 1993).

As a final point we suspect that the ability to navigate within a common spatial frame and to be aware of the presence and location of others requires a sufficiently large space. It may well be that 2-D interfaces on a workstation screen may be too small for such an approach to be possible. We believe that the long term solution to this problem is to move from 2-D to 2.5-D or 3-D interfaces (the former being the monoscopic pseudo-3D interfaces used in “desktop virtual reality” and the latter being stereoscopic 3-D displays use in immersive virtual reality). Until you have experienced such an interface, it is hard to appreciate just how much additional space is provided through the addition of a third dimension and just how much this opens up the possibility for peripheral awareness of others. Indeed, virtual reality systems seem to automatically embody users within a common space, providing representations of both viewpoints and actionpoints. Thus, we argue that future collaborative interfaces will move away from the current two dimensional approaches inherent in WYSIWIS and video towards more sophisticated use of multi-dimensional space.

3.5. Summary

Support for cooperative work requires an appropriate degree of synchronization of user interfaces. Too little synchronization and users have no common ground; too much and coordination becomes problematic. An analysis of two key CSCW technologies, WYSIWIS interfaces and video, identified a number of key limitations. More specifically, existing approaches fail to distinguish between individual viewpoints, actionpoints and having a common spatial frame of reference and also fail to adequately embody users within the common display space. The chapter then proposed a framework for the design of collaborative user interfaces which aims to overcome these problems. This framework identifies the following key components: selection, presentation, spatial frame, embodiment, viewpoints and actionpoints.

Considering synchronization, the chapter argued that the spatial frame needs to be synchronized but viewpoints and actionpoints do not. Thus, users are free to move autonomously within a common space. They should also be aware of the presence and movement of others. It also proposed that viewpoints and actionpoints need not be strictly coupled together for an individual. Furthermore, different synchronization policies could be applied to separate sub-spaces through a process of unfolding. As a final note, the chapter suggested that two dimensional interfaces are too cramped to realize these ideas in a general way and that a move towards three dimensional interfaces is required if collaborative work is to be supported in a flexible and natural manner.

This completes the review phase of the COMIC spatial work. The following chapter begins the development of a COMIC “spatial model” of interaction which attempts to meet the goals laid down by the above design framework and also addresses the issues raised in chapter one and the requirements identified by chapter two.

4. A Spatial Model of Interaction in Large Virtual Environments

Steve Benford and Adrian Bullock
University of Nottingham

John Bowers
University of Manchester

Lennart Fahlén
SICS

This chapter presents a spatial model of group interaction in large virtual environments which aims to meet the requirements identified in chapters one to three. The model provides flexible support for managing conversations in virtual space. It can also be used to control more general interactions among other kinds of objects inhabiting such spaces. The key abstractions of aura, awareness, nimbus, focus adapters and boundaries are defined in a sufficiently general way so as to apply to a wide range of CSCW systems.

The chapter describes several possible and emerging applications of the spatial model including its application to virtual reality, text conferencing and information spaces (the latter resulting in a concept called Populated Information Terrains). The chapter then analyses the model in relation to previous studies of cooperative work in CSCW. More specifically, this analysis shows how the model can support a number of awareness states which have been shown to be important to cooperative work within the real world. Following this, we explore the relationship between the model and emerging work on Open Distributed Processing. Finally, the chapter ends by identifying several unresolved issues for future study.

4.1. Introduction

This chapter presents an initial model for supporting group interaction in large-scale virtual worlds that aims to meet some of the requirements identified in chapters one to three. In particular, it focuses on support for interaction at the "microscopic level" (i.e. supporting conversation management within a local space) as discussed within chapter two. It also supports the notion of separating a common spatial frame from the viewpoints and actionpoints of embodied individuals as proposed by chapter three.

The model provides generic mechanisms for managing interactions between various objects in such spaces including humans and computer artefacts.

Furthermore, the model is intended to be sufficiently flexible to apply to any system where a spatial metric can be identified (i.e. a way of measuring distance and orientation). Such applications might range from the obvious example of multi-user virtual reality through conferencing systems, collaborative hypermedia and even databases and information spaces.

Where the interacting objects are humans, the model provides mechanisms for conversation management. These contrast with existing floor control and workflow modeling techniques by adopting a "spatial" approach where people employ the affordances of virtual computer space as a means of control. In so doing, our underlying philosophy has been to encourage individual autonomy of action, freedom to communicate and minimal hard-wired computer constraints. Where the interacting objects are artefacts, the model provides mechanisms for constructing highly reactive environments where objects dynamically react to the presence of others (e.g. you may activate a tool simply by approaching it or even glancing at it).

Section 4.2 introduces the key concepts belonging to the model — aura, focus, awareness, nimbus, adapters and boundaries. Section 4.3 discusses possible applications of the model and describes some early concept demonstrators. Section 4.4 introduces some diagrammatic notations for the model and uses these to relate the model to important awareness phenomenon observed within previous studies of cooperative work in the "real world". Section 4.5 explores the relationship between the spatial model and emerging models of distributed processing, particularly the notion of trading within distributed systems. Finally, section 4.6 summarises the chapter and identifies some key research issues for future consideration.

4.2. Key concepts in the spatial model

The spatial model, as its name suggests, uses the properties of space as the basis for mediating interaction. Thus, objects can navigate space in order to form dynamic sub-groups and manage conversations within these sub-groups. We introduce the key abstractions of **SPACE**, **OBJECTS**, **MEDIA**, **AURA**, **AWARENESS**, **FOCUS**, **NIMBUS**, **ADAPTERS** and **BOUNDARIES** which define the model.

Space and objects

The most fundamental concept in our model is **space** itself. Michael Benedikt in his paper "Cyberspace: First Steps" defines space as "freedom to move" and proceeds to discuss the structure of virtual space in terms of combinations of dimensions which might behave in continuous or discrete ways (Benedikt, 1991). Following this line, we also allow virtual spaces to have any number of dimensions where each dimension allows some measure of position. Put another way, space is defined by 'spatial metrics' — well defined ways of measuring position and direction across a set of dimensions (we return to this issue in section 4.3 when discussing Populated Information Terrains).

Space is inhabited by **objects** which might represent people, information or other computer artefacts. Any interaction between objects occurs through some **medium.** A medium might represent a typical communication medium (e.g. audio, visual or text) or perhaps some other kind of object specific interface. Each object might be capable of interacting through a combination of media/interfaces and objects may negotiate compatible media whenever they meet in space.

Aura

The first problem in any large-scale environment is determining which objects are capable of interacting with which others at a given time. Two problems will emerge as spaces become densely populated:

1. Cognitive overload — it will not be feasible for every inhabitant of space to perceive every action of every other inhabitant.
2. Computational overload — it is not computationally scaleable to "transmit" every action from each object to every other in a crowded space (the number of transmitted messages explodes exponentially of the order 2^n where n is the number of objects in the space).

Some method is required of localising interaction to a given spatial area in order to reduce these problems of overload. **Aura** is defined to be a sub-space which effectively bounds the presence of an object within a given medium and which acts as an enabler of potential interaction (Fahlén, 1992). Objects carry their auras with them when they move through space and when two auras collide, interaction between the objects in the medium becomes a possibility. It is the surrounding environment that monitors for aura collisions between objects. When such collisions occur, the environment takes the necessary steps to put the objects in contact with one another (e.g. exchange of object IDs, addresses, references or establishment of associations or connections). Thus, aura acts as a fundamental technological enabler of interaction and is the most elementary way of identifying a subspace associated with an object. An aura can have any shape and size and need not be around the object whose aura it is. Nor need it be contiguous in space. Also, each object will typically possess different auras for different media (e.g. with different sizes and shapes). For example, as I approach you across a space, you may be able to see me before you can hear me because my visual aura is larger than my audio aura.

Awareness, focus and nimbus

Once aura has been used to determine the potential for object interactions, the objects themselves are subsequently responsible for controlling these interactions. This is achieved on the basis of quantifiable levels of **awareness** between them (Benford, 1992). The measure of awareness between two objects need not be mutually symmetrical. As with aura, awareness levels are medium specific. Awareness between objects in a given medium is manipulated via **focus** and **nimbus**, further subspaces within which an object chooses to direct either its presence or its attention. More specifically, if you are an object in space:

- The more an object is within your focus, the more aware you are of it.
- The more an object is within your nimbus, the more aware it is of you.

The notion of spatial focus as a way of directing attention and hence filtering information is intuitively obvious from our everyday experience (e.g. the concept of a visual focus) and has been explored in previous work on user interface design (e.g. fish-eye views (Furnas, 1986)). The notion of nimbus requires a little more explanation. In general terms, a nimbus is a sub-space in which an object makes some aspect of itself available to others. This could be its presence, identity, activity or some combination of these. Nimbus allows objects to try to influence others (i.e. to somehow project themselves or "be heard"). Thus, nimbus is the necessary converse of focus required to achieve the desired power balance in interaction.

Objects negotiate levels of awareness by using their foci and nimbi in order to try to make others more aware of them or to make themselves more aware of others. We deliberately use the word negotiate to convey an image of objects positioning themselves in space in much the same way as people mingle in a room or jostle to get access to some physical resource. Awareness levels are calculated from a combination of nimbus and focus. More specifically, given that interaction has first been enabled through aura,

> *The level of awareness that object A has of object B in medium M is some function of A's focus on B in M and B's nimbus on A in M.*

The resulting quantified awareness levels between two objects can then used as the basis for managing their interaction. Exactly how this is achieved is a matter of interpretation by a particular application. One approach might be to use awareness levels to directly control the medium (e.g. controlling the volume of an audio channel between two objects). Another might be allowing objects to actively react to each others presence depending on specified awareness thresholds (e.g. I might automatically receive text messages from you once a certain threshold had been passed). Several example applications are discussed later.

Notice that aura and then focus/nimbus are layered concepts. First, aura is applied at the distributed systems level to deal with the problem of computational overload. Only after aura collision has occurred and the potential for interaction been established, will users be able to apply focus and nimbus to manage the interaction.

Manipulation and representation of aura, focus and nimbus

Next we consider how aura, focus and nimbus, and hence awareness are displayed to and manipulated by users. The spatial model aims to provide a natural and flexible approach. This requires that users are generally unaware of its detailed operation and that key concepts such as focus and nimbus are conveyed and manipulated in implicit rather than explicit ways. Aura is an underlying technical enabler of communication and users should be unaware of its presence and effect. Focus and nimbus are more user-oriented concepts. Their presence should be conveyed in natural ways (e.g. the use of appropriate body images to convey position, orientation and factors such as "gaze" location in others). We envisage three primary ways of users manipulating aura, focus and nimbus:

1. Aura, focus and nimbus may most often be **implicitly** manipulated through fundamental spatial actions such as movement and orientation. Thus, as I move or turn, my aura, focus and nimbus might automatically follow me. A number of novel interface devices are emerging to support this kind of movement. These are generally known as six dimensional devices (three for position and three for orientation) and include space-balls, body-trackers, wands and gloves.
2. They may on occasion be **explicitly** manipulated through a few key parameters. A user interface might provide a few simple parameters to change aura, focus and nimbus. For example, I might change the shape of a focus by focusing in or out (i.e. changing a focal length). This might be achieved by simply moving a mouse or joystick.
3. They may be manipulated through the action of various **adapter objects** and **boundary objects** as described in the following section.

Adapters and boundaries

Aura, focus and nimbus may be manipulated through various other objects which modify them in some way. We group such objects into two categories, **adapter objects** and **boundary objects**. Adapters might be represented in terms of natural metaphors such as picking up a tool and support interaction styles beyond basic mingling. In essence, an adapter is an object which, when picked up, amplifies or attenuates aura, focus or nimbus. For example, a user might conceive of picking up a "microphone". In terms of the spatial model, a microphone adapter object would then amplify their audio aura and nimbus As a second example, the user might sit at a virtual "table". Behind the scenes, an adapter object would fold their aura, foci and nimbi for several media into a common space with other people already seated at the table, thus allowing a semi-private discussion within in a space. In effect, the introduction of adapter objects provides for a more extensible model.

Focus, aura and nimbus may be manipulated through **boundaries** in space. Boundaries can be thought of as having (at least) four kinds of effects:

- effects on aura;
- effects on nimbus;
- effects on focus;
- effects on traversal.

These effects can be of one of four sorts in relation to the kind of boundary in question — obstructive, non-obstructive, conditionally obstructive and transforming. Obstructive means that the boundary blocks the property in question (movement, aura, focus or nimbus). Non-obstructive means that it does not. Conditionally obstructive means that the obstruction can be removed when some condition is met (e.g. possession of a "key"). Finally, transforming means that the boundary alters the property in some way. Each of these effects might be defined separately on a per medium basis and also separately for the direction of traversal (some boundaries are obstructive one way but not the other.). We identify several different general kinds of boundary which might emerge under this model:

- Permeable boundaries. These are not obstructive. One's aura etc. is not diminished in crossing the boundary. Traversal is not effected ("free passage").
- Semi-permeable boundaries. These come in two sorts. There are those which *transform* aura and the rest which typically diminish it (though in a virtual world, other transformations can be conceived). There are boundaries which resist the extent to which, say, focus can pass through (frosted glass) and there are boundaries which resist traversal (ditches). Also, there are semi-permeable boundaries for which the passage of aura etc. or traversal might be *conditional* on assistance from an adapter or a means of transportation.
- Impermeable boundaries. These are boundaries which *no* assistance from an adapter or a means of transportation will make permeable. Of course, such boundaries are not to be regarded as for always and ever impermeable. Rather, the resources that one requires for making them so are of a different sort from amplifiers and means of transport. You have to demolish and/or rebuild such boundaries to make them permeable or semi-permeable. (It is worth noting that in the so-called real world, demolition is rarely a silent, private act and is often explosive! In the terms of this chapter, the assembly made of a wall (say) with explosives and lighted fuses at the point of ignition has very large aura and nimbus! In a virtual world for cooperative work, this may sound sensible: fundamental changes in the spatial architecture get broadcast.)

Let us briefly digress and relate this concept of boundaries within the spatial model to the discussion of boundaries in chapter two. In an approximation of Bentham's panoptical prison in our terms, three of each cells' walls were impermeable to focus, nimbus and traversal. The fourth is perhaps unusual in that it was permeable to the observer's focus and the prisoner's nimbus, yet impermeable to the observer's nimbus and the prisoner's focus (like a one way mirror). All four walls are impermeable to the prisoner's traversal (we suggest that it is the residual collision of auras in such an asymmetric architecture which Bentham hoped would make the prisoner self-aware). While there are one way mirrors and subtly designed apertures in prison architecture, it is hard to find in the so-called real world a boundary which is permeable to nimbus and impermeable to focus but permeable to traversal! (Mirrors in some myths and children's stories afford traversal but we do not know of a myth concerning a one way mirror that you can step through — but we're prepared to be corrected!)

It is worth exploring the relationship between boundaries and adapters a little further. Any object can act as a boundary and, although one tends to think of boundaries in terms of walls, doors and windows, we are all familiar with the fact that furniture and even people can make very effective boundaries (e.g. a well placed desk or a row of policemen). Thus, although it may help to explain the model if we conceive of boundaries and adapters as being different kinds of object, they are really the same thing. In fact, at the most abstract level, all objects can serve to obstruct, non-obstruct, conditionally obstruct or transform aura, nimbus and focus. In other words, we are really identifying fundamental properties of all objects in space and their effects on the spatial model. Modeling such a situation may require a rather more powerful underlying formalism such as some kind of "field theory" (consider the analogous relationship between real-world objects and gravitational fields). The identification and use of such a formalism is an issue for future work.

To summarise, our spatial model defines key concepts for allowing objects to establish and subsequently control interactions across virtual space. First, we allow an abstract and flexible definition of multi-dimensional space. Aura is used to establish the potential for interaction across a given medium. Focus and nimbus are then used to negotiate the mutual and possibly non-symmetrical levels of awareness between two objects which in turn drives the behaviour of the interactions. Finally, objects can act as adapters and boundaries and can further influence aura, focus and nimbus thereby adding a degree of extendibility to the model. The following section now describes a number of existing and proposed applications of the spatial model.

4.3. Applying the spatial model

The spatial model is intended to be applicable to any system where a spatial metric can be identified. This section describes some example applications of the spatial model to a range of systems including virtual reality, text conferencing and information and hypermedia spaces. The discussion includes a number of screen shots from early concept demonstrators which have been developed as a pre-cursor to later prototyping activities within COMIC.

Sections 4.3.1. and 4.3.2. focus on the application of the model to Virtual Reality systems. Section 4.3.1. discusses the realisation of the aura concept within the DIVE (DIstributed Virtual Environment) developed at SICS. Section 4.3.2. describes some focus and nimbus concept demonstrators developed at Nottingham. Finally, section 4.3.3 identifies the application of the model to more abstract information spaces, resulting in the notion of a *Populated Information Terrain*.

4.3.1. Implementing aura within the DIVE system

Perhaps the most obvious application of the spatial model is to virtual reality systems. A prototype multi-user Virtual Reality (VR) system, DIVE (Distributed Interactive Virtual Environment) (Fahlén, 1991) (Carlsson, 1992) has been developed as part of the MultiG program (a Swedish national research effort on high speed networks and distributed applications (Pehrson, 1992)). DIVE is a UNIX-based, multi-platform software framework for creating multi-user, multi-application, three-dimensional distributed user environments. There is support for multiple co-existing "worlds" with gateways between them to enable inter-world movement.

Users are represented by unique graphical 3D-bodies or icons whose position, orientation, movements and identity are easily visible to other participants. In this first realisation, aura is implemented as a volume or sphere around each user's icon which is usually invisible. Aura handling is achieved through a special collision manager process. When a collision between auras occurs, this manager sends a message containing information such as the id's of the objects involved, positions, angles and so on, to other processes within the DIVE environment. These processes (e.g. the owners of the objects involved) then carry out appropriate focus, nimbus and awareness computations. It is possible to have support for a multiple users, objects, media and service specific aura types with associated collision managers mapped onto separate processing nodes in a network. Focus and nimbus handling can be mapped in a similar way. Further details on the aura implementation in DIVE can be found in (Ståhl 1992b). Figure 4.1 shows a screen dump from DIVE of an aura collision, with the auras made specially visible.

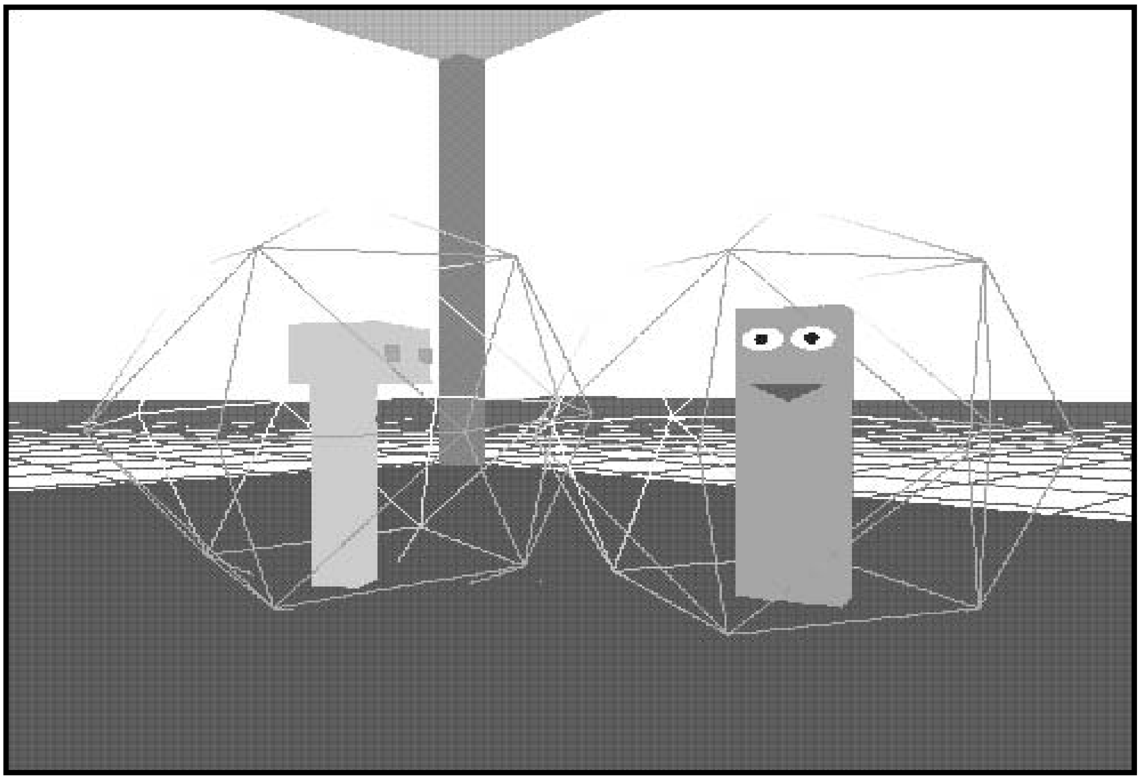


Figure 4.1: Body Images with Colliding Auras

A more general tool kit has been developed as a first step towards constructing a distributed collaborative environment and for experimentation with the concepts of aura, focus, nimbus and awareness. This tool kit has been used to develop a number of adapter and other objects within DIVE.

• The whiteboard

The whiteboard (Ståhl 1992a) is a drawing tool similar in appearance to it's real world counterpart. Several users can work together simultaneously around the whiteboard. There can also be groups of whiteboards, with the contents being duplicated across the group. That is, the actions performed by one user on one whiteboard are immediately replicated by the other whiteboards in the same group. Each whiteboard has its own aura which is used to enable whiteboard access and use (e.g. by automatically assigning a pen to a user when their aura collides with that of the whiteboard). In this case we are seeing the use of spatial model in managing interaction between humans and non-human objects. The result is that the non-human objects become reactive, that is they react to the presence or even gaze of people.

• Documents

The content of a whiteboard can be copied into something called a document that a user can pick up and carry away. Apart from being "single user", documents have the same functionality as a whiteboard. More specifically, when document auras intersect, their contents are copied to other users documents and onto whiteboards. In this case we are seeing the spatial model being used to drive interactions between non-human objects.

• The conference table adapter object

The conference table is an adapter object which detects participants presence, and establishes communication channels (video, voice and document links) between them via aura. The auras, foci and nimbi of the conference participants around the table are then extended to cover everyone in attendance. So, by having a very long table, people can form larger collaborative groups than "direct" aura/focus/nimbus functionality makes possible. Users can come and go as they please and it is easy to obtain an overview of who is present. The conference table can also distribute documents to conference participants and to whiteboards. To do this a user simply places a document in the centre of the table and then the aura collision manager initiates the distribution. Figure 4.2 shows a screen dump of a meeting in Cyberspace involving the whiteboard and conference table.

The podium adapter object

A participant can enter a podium adapter object and is thereby allowed to be "heard" (or seen) by a dispersed group of users that "normally" (e.g. without the podium) are not within communication distance of the "speaker". The aura and nimbus of the participant on the podium are enlarged to cover, for example, a lecture hall or town square. The podium is an example of an aura/nimbus adapter and it is asymmetric, i.e. the "listeners" can't individually communicate back to the "speaker" without special provisions.

A teleconferencing subsystem is also under construction and will be integrated into DIVE in the near future (Eriksson, 1992) . Apart from the CSCW tool kit, some other concept demonstrators have also been developed within the DIVE environment, including control of a real-world robot, a 3D customisable graph editor for drawing and editing graphs in 3D space, a 3D-sound renderer allowing objects or events to have sounds and for these sounds to have a position and direction and finally a computer network visualiser and surveillance tool. Technically, DIVE is not based on a client-server model, instead all participating nodes are treated as peers with respect to control and data distribution. This distribution is presently implemented by use of the ISIS distributed programming tool kit (Birman, 1991).

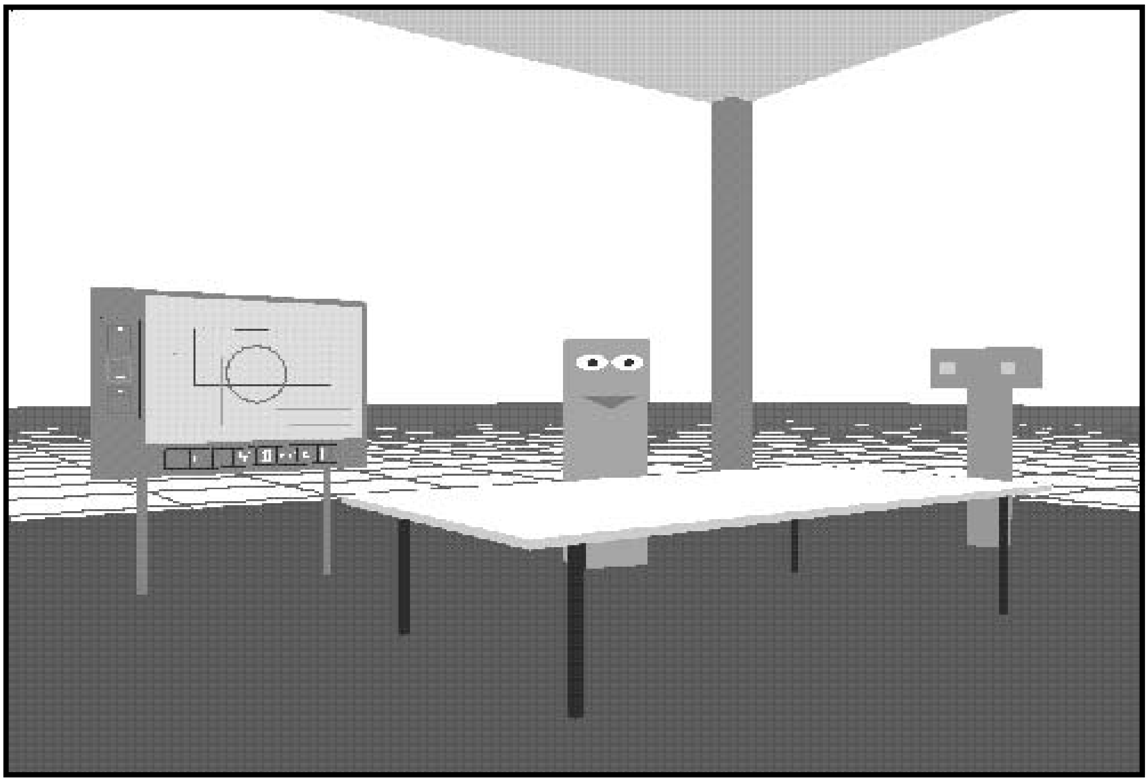


Figure 4.2: A Conference in Cyberspace

4.3.2. Realising focus and nimbus within "virtual reality"

This section describes a number of concept demonstrators which have been produced at Nottingham in order to more clearly communicate concepts of focus and nimbus. The demonstrators are built on a PC based system called Superscape[1]. Three demonstrators will be described, showing the concepts of focus, then nimbus and then a combination of both. The first demonstrator shows a possible bulletin board interface where different people post articles and periodically inspect the board to see if any new articles have arrived. This demonstrates the concept of focus and also the notion of reactive objects. The second demonstrator illustrates the

1 Superscape is a product of Dimension International

use of both nimbus and adapters to influence awareness. The demonstrator allows you to enter a crowded space where many people are randomly walking around and to attract their attention in various ways. The third demonstrator illustrates the combination of focus and nimbus. We enter a space of abstract objects. The level of detail shown depends on our awareness of the object in question which, in turn, is governed by our focus (sensitive to position and orientation) and to the object's nimbus.

First a note on embodiment. Within the three demonstrators people are represented in two minimalist ways. Some people are embodied as rectangular boxes with a facial design on the front to denote direction (affectionately known as filing cabinets). These body images are based upon some of those from the DIVE system described above. Second, people maybe represented as "snouts", a pyramid based shape which allows direction and orientation to be deduced by simple observation. Several screen dumps are included to give a flavour of the demonstrations. However, it is recommended to experience the actual demonstrations to get a real feel of the concepts of focus, nimbus, awareness and adapters.

The Bulletin Board — a focus demonstrator

The bulletin board presents the user with a screen like representation of a series of bulletin boards onto which notices have been posted. From a distance they see only the boards with a numerical representation relating to the number of articles or groups of articles on the boards. In this minimal awareness case, they see a simple summary of the state of the board (in the case of figure 4.3, six new articles have appeared since they last looked).

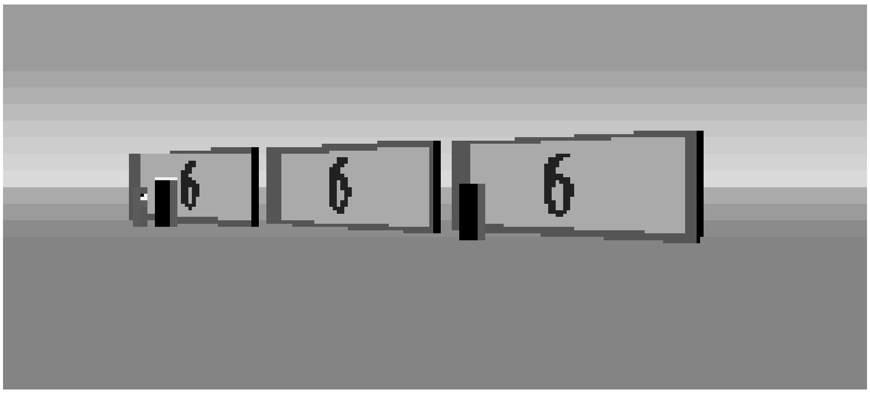


Figure 4.3 — minimal awareness of the bulletin boards

As they move closer to the board they see these numbers disappear to be replaced by minimal representations of the articles (figure 4.4). Colour is used to represent different groupings of articles (not really visible in this monochrome screen dump). In this case, previously read articles are one colour and new articles another. This affords an instant overview of where to direct their attention. Also, as the board is located in a shared space they can also see other people who are reading the board, and in particular those who are reading the same section of the board. This introduces the opportunity for possible collaboration.

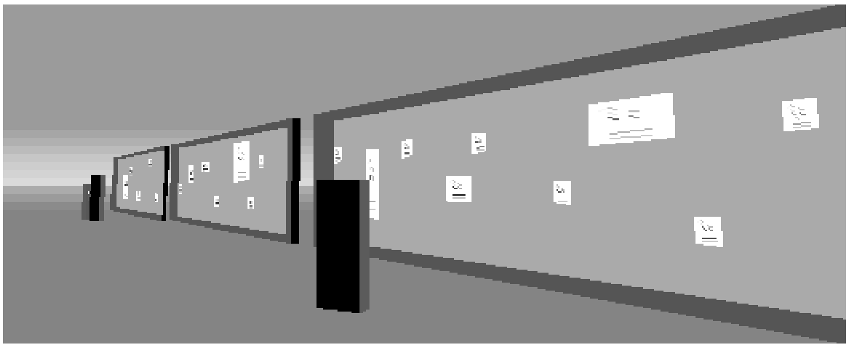


Figure 4.4: Awareness of where interesting articles are located

At this stage they see overall patterns of information, but no text about the subjects of the articles. They have to increase our awareness of the board to see this by moving closer (Figure 4.5). When they are close enough to an article its subject header is displayed (and maybe other summary information such as an image of the author).

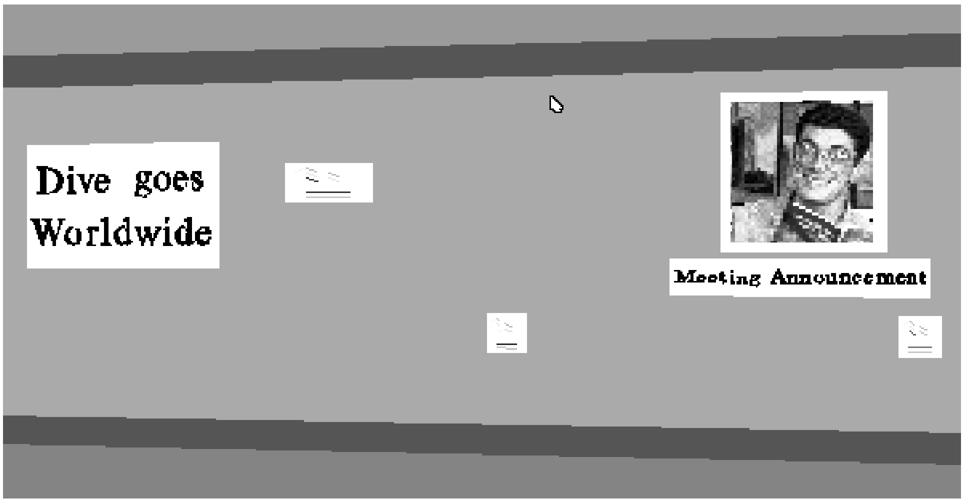


Figure 4.5: Increasing awareness produces summary information

By moving even closer still, individual articles are displayed (Figure 4.6) and they can get down to the task of reading them. This is achieved by the text of the article appearing at the base of the screen and the headline of the article being highlighted to show it is currently being read. Note that the readers focus is directional. Thus, if they shift their glance sideways, they become more aware of the things they are directly looking at, and less aware of the articles they were previously looking at.

Figure 4.6: Maximum awareness of the board — reading a notice

The Podium — a nimbus demonstrator

The scenario in the second demonstrator is a space in which there are many people mingling randomly, with a podium at the centre of the space. The inhabitants of the space are happily moving around, bumping into other people as they go, being unaware of what is happening outside of their local surroundings. There is a special person within this crowd, a VIP say, who, when the rest of the inhabitants pass close enough to her, they are ‘caught’ by her presence. That is she has a local nimbus which makes others who pass within it aware of her (the position of the VIP is shown as a small square). This awareness is represented in the prototype by making the other people stop, and turn to face the VIP (Figure 4.7). They remain interested in the VIP for a random amount of time but eventually get bored of what she is saying and manage to ‘break-away’ from her nimbus and resume their random mingling about the space.

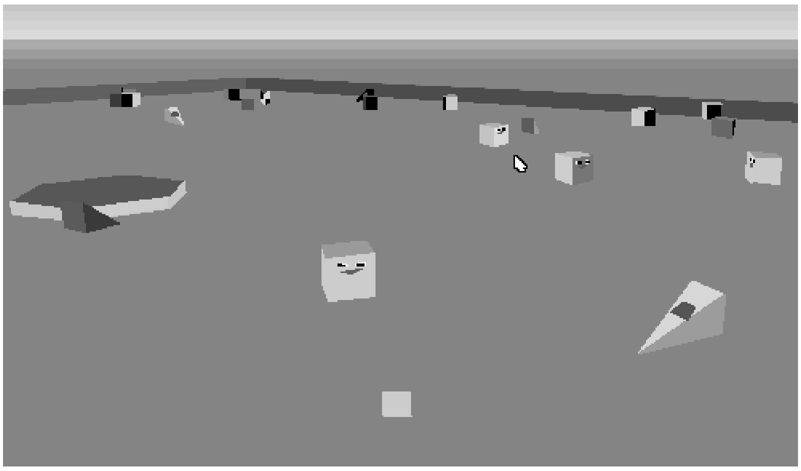


Figure 4.7: A “VIP” with personal nimbus mingles in a crowded space

At some stage the VIP wishes to address the whole of the crowd. This could be likened to an address at a conference or maybe a soloist about to perform on stage. To do this, she needs some way of projecting her nimbus over everyone in the space. This is where the podium comes in. The podium acts as a nimbus amplifier, that is it projects the nimbus of the person standing on it over the whole of the space. When our VIP steps onto the podium everybody in the space stops moving immediately, and over the next few seconds turn to face her (Figure 4.8).

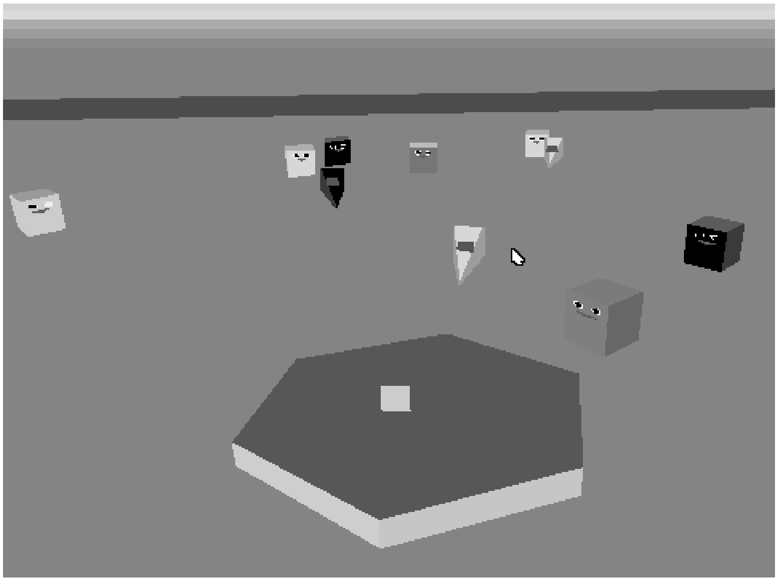


Figure 4.8: Our VIP uses a nimbus amplifier to address the crowd

When she steps back off of the podium, everyone else in the space returns to their random mingling, save those who are close enough to the VIP to still be within her own personal nimbus.

The stairs — a demonstrator of both focus and nimbus

The third and final demonstrator shows how a combination of focus and nimbus influence awareness. The demonstration space contains a number of objects clustered around a cross-shaped staircase (Figure 4.9). Some of the objects are on the ground whilst others are on the staircase itself. There is a single person occupying the space whose viewpoint we take in the screen-dumps. Our awareness of each object is mapped onto the level of detail shown. Thus, minimal awareness results in seeing a simple block outline of the object and maximal awareness in seeing a double-tapered and fluted column (intermediate levels result in intermediate representations). More specifically, the observer's level of awareness of an object is influenced by *their focus* which is sensitive to both distance and orientation and by the *size of the object's nimbus*. Thus, some of the more distant objects can be seen in their full detail whilst other objects can only be seen in only partial detail even though they are closer.

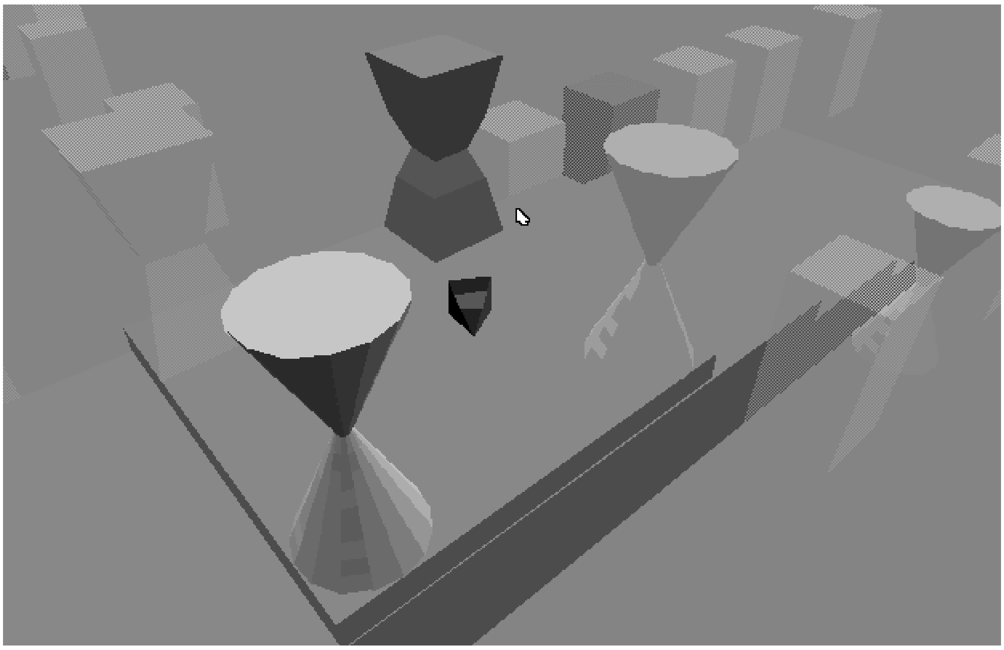


Figure 4.9: detail depends on the object's nimbus and the observer's focus

4.3.3. Populated Information Terrains

The applications of the model discussed above have been oriented towards realistic looking spaces such as meeting rooms which contain recognisable objects from the real-world such as tables, white-boards, notice-boards and podia. This section proposes an alternative application of the model that is less brutally real and far more abstract in its approach. The approach centres on the notion of constructing multi-dimensional data spaces to represent large collections of data (i.e. database visualisation). The data might be a collections of documents, records, measured values or other objects. We refer to such a space as an *Information Terrain*.. Once constructed, information terrains can be populated with people who might communicate across and through the information. This communication might be mediated by the spatial model. The spatial model requires the construction of a space according to some “spatial metric”. In terms of information terrains, this means finding techniques for mapping collections of data into appropriate spatial structures. A wide variety of approaches are possible, several having been discussed in previous literature. The remainder of this section outlines several of these approaches towards constructing such *Populated Information Terrains*. We anticipate that later COMIC work will implement some of these and will apoly the spatial model to mediate interaction within the resulting spaces.

Multi-dimensional visualisation

When discussing the dimensionality of cyberspace Michael Benedikt explores the problem of directly mapping attributes of objects onto sets of spatial dimensions (Benedikt 1991). His approach is to select a few attributes (typically three) which can be used to plot the position of an object in space after suitable scaling. The chosen attributes are said to map onto the object’s *extrinsic dimension*s. The remaining attributes are mapped onto characteristics of the resulting point in space such as colour, size, spin, vibration and sound. These are referred to as *intrinsic dimensions*. Presumably, the user can dynamically (re)select the extrinsic and intrinsic dimensions and so (re)construct the space at will. However, Benedikt also proposes the principles of Exclusion and Maximal Exclusion which guide the choice of extrinsic and intrinsic dimensions. The Principle of Exclusion states that:

> “Two non-identical objects having the same extrinsic dimensions and dimension values, whether at the same time, or including time as an extrinsic dimension from the outset, is forbidden, no matter what other comparisons may be made between their intrinsic dimensions and values.”

In other words, you cannot have two different things in the same place at the same time. The Principle of Maximal Exclusion states that:

> “Given any N-dimensional state of a phenomenon, and all the values — actual and possible — on those N-dimensions, choose as extrinsic dimensions — as “space and time” — that set of (two, or three, of four) dimensions that will minimise the number of violations of the Principle of Exclusion.”

In other words, chose your dimensions so that different objects are spatially separated as much as possible. This avoids the information loss that occurs when multiple objects collapse onto a single point of space. Benedikt’s approach is therefore to draw a graph of the data (usually in 3-D) and to guide the selection of the co-ordinate system. We can relate Benedikt’s Principle of Maximal Exclusion to the notion of a relational database key. To be more precise, if the attributes that comprise the key for an object are included in those mapped onto extrinsic dimensions, then Benedikt’s principle will be satisfied.

Statistical clustering and proximity measures

A second approach to constructing information terrains is provided by statistical methods which analyse collections of data (often documents) in an attempt to group objects together according to some measure of semantic "closeness" (i.e. do they logically belong together). The resulting proximity measures are typically scaled and returned as numerical values which are then used to locate the objects in space. We briefly mention three examples.

VIBE is an experimental visualisation interface for information retrieval systems (Olsen 1993). The system allows a single user to specify a number of *points of interest (POI)* — keyword search queries which are associated with a particular spatial position. These POIs might represent queries, interest profiles or even positions of well known documents and serve to frame the visualisation space. The system then conducts a full text search on the information base, matching each document against each POI. The result is that each document becomes associated with a vector of which each element denotes a score (closeness of match to the keyword pattern) and position (of the POI). This vector is then used to position the document in the visualisation space on the basis of a centre of gravity between all the POI positions, weighted by the scores. The advantage of the VIBE approach is that users can specify as many POIs as they like in whatever positions they like and can dynamically re-frame the space in order to improve the view by defining new POIs or moving existing ones.

Another approach was used in the BEAD system and subsequent work on document landscapes (Chalmers 1992). BEAD employs a range of numerical techniques combined with text analysis of keywords in academic papers to construct three dimensional point clouds in space where each point represents an individual document. Later work has revised the point cloud interface to that of a landscape which attempts to make the document space more navigable. The idea here is that people are better able to navigate a landscape from an aerial position than they are a fully three dimensional structure. This is based on the observation that our experience of the real world is not fully three dimensional (we tend to treat the vertical dimension as a special case). Thus, Chalmers argues for a less than full 3-D information space (he terms his approach "2.1 D").

A related approach was followed by Callon who attempted the systematic mapping of scientific disciplines (Callon). As with the BEAD system, this work performed a statistical analysis of the keywords of papers in an attempt to deduce relationships between them. This time two measures or indices were calculated: *inclusion*, a measure of how much one keyword is subsumed by another, and *proximity*, a measure of how close in meaning two keywords are. The resulting indices where used to construct two dimensional maps of the status of scientific disciplines as they evolved by relating commonly used terms (keywords) to each other in a spatial structure.

TripleSpace

A three dimensional technique for database visualisation, TripleSpace, is presented in (Mariani 1992). This system is built on top of a triple store or binary relational database system. All information within such a system is represented as a triple, which can generically be described as (subject, relation, object). These “three field records” present themselves readily to the prospect of being represented as (x, y, z) co-ordinates in space. While Benedikt’s approach, and that of (Young 1991), show the need to choose three out of N field domains to convert to the (x, y, z) co-ordinates, and hence provide a mapping from the N-dimensional information space to the 3-D visual space, TripleSpace equates the information space directly to the visual space. A point in TripleSpace is represented as a cube. The sides of the cube normally contain letters representing the subject, relation and object of the triple, although it is possible to associate an icon with an item of data and this will appear instead of text on a cube face. TripleSpace illustrates the potential of data visualisation by allowing the user to uncover interesting cases of data (by visual inspection of clusters) that would not normally be uncovered by an inspection of the textual data. Three examples of such cases are given in (Mariani 1992). The user is free to navigate through the TripleSpace and explore the data topology.

Hypertext and spanning trees

Many information systems employ the notion of directly linked objects including hypertext, hypermedia and Network Information Retrieval (NIR) systems such as Gopher, ARCHIE, WAIS and World Wide Web. In particular, emerging NIR systems provide access to a global space of linked information objects. In these systems, a measure of spatial distance can be taken as the shortest path number of links between objects. An alternative approach to visualising such a structure might be to draw three dimensional connected graphs or minimal spanning trees emanating from some starting node in the structure. For example, the nodes of a minimal spanning tree might be arranged on a series of concentric spheres with all those one-link away being on the inner, those two links away on the next and so on. This approach might be enhanced by the application of *fish eye view* and *cone tree* techniques for managing perspective views of large volumes of information (Furnas 86).

Employing real world metaphors

So far, we have focused on techniques for automatically constructing and visualising Information Terrains. By way of contrast, we now consider two human centred approaches which rely on users or system implementors designing appropriate visualisations. The first of these is the use of real-world metaphors to represent information terrains. Examples might include: constructing a three dimensional fly-through *library* interface to represent a space of documents; using the metaphor of *cities, buildings* and *rooms* to visualise large databases of organisational information such a Directory services and name servers; and using *maps* of the physical world as a basis for displaying environmental data (effectively what happens with

Geographical Information Systems). The advantage gained from such metaphors might be greater familiarity for users and hence increased usability. Of course, the disadvantage is that such metaphors may also import constraints from the real-world which would be unnecessary in computer implementations.

Human structured spaces

Our final approach is to allow humans to construct and organise the Information Terrain themselves on an ad-hoc basis. We are all familiar with this approach from real-world desktops which often contain piles of related documents arranged by their owners according to some personal strategy. These piles can be reshuffled, repositioned and restructured at will to provide convenient access to information. The desktop metaphor has been employed very successfully in interfaces to computer filestores. Such interfaces allow users to construct and configure their own information terrains based on hierarchical nesting of files within directories. We might extend this approach to larger-scale three dimensional Information Terrains. Why not allow users to construct piles or clusters of data in three-dimensional space according to some personal or group classification scheme? In other words, why not let people place or throw the data where they want?

This concludes our discussion of approaches for constructing Populated Information Terrains and indeed, of possible applications of the spatial model. The next section turns its attention towards the relationship between the spatial model and various awareness phenomena which have been observed in previous studies of cooperative work.

4.4. Relating the model to studies of cooperative work

This section attempts a further explanation and interpretation of the spatial model in the light of previous observations of the importance of awareness in cooperative work. The aim of this analysis is to show how the model fits with and supports key features of cooperative work observed by other researchers in the field. Thus, this sections serves as both a clarification and also a further justification of our model.

The 'mechanism' of the COMIC spatial model (i.e. how the concepts inter-relate) enables the modeling (indeed, the calculation) of mutual awareness between people. *Exactly what content is given to these distinctions is a matter of interpretation*. It is here that an examination of the social scientific studies of spatiality and awareness mentioned in chapter two might be of assistance. Fundamental to much of the work examined there was a notion of *making one's conduct available to others*. It is this that, for present purposes, we shall interpret as a useful, more concretised meaning for nimbus. On this interpretation, *one's nimbus is a subspace in which one project one's activities*. Drawing similar inspiration from the empirical studies mentioned earlier, we can interpret one's focus as: *a subspace in which one attends to the activity of some entity*. It seems reasonable to suggest an interpretation for aura given these interpretations of nimbus and focus. The collision of auras is the precondition of the subsequent 'levels'. In an approximate sense, awareness of the *presence* of an entity is presupposed by attending to its activity. This view suggests, then, that *an entity's aura is a subspace in which it projects its presence*. (Naturally, other interpretations are possible and — for the general development of the spatial model — specific interpretations should be resisted. However, concretising the terms in this way is motivated and enables us to retain some contact with empirical studies of work practice.

Let us work through a specific instantiation of the spatial model a little further under the interpretation we have given so far. We want to analyse a simple, discrete model involving just two entities, call them A and B. The model is discrete in the sense that the subspaces of aura, nimbus and focus are well-bounded. Any specific location in this discrete model will, then, be unambiguously either inside or outside a subspace of aura, nimbus or focus. **Please note that this represents a major simplification of the concepts of nimbus and focus**. In what follows, we are going to analyse the different modes of awareness that such an instantiation of the spatial model might recognise. As, on the interpretation of awareness we are pursuing, any awareness presupposes a minimal awareness of presence, we are going to simplify the analysis by only considering focus/nimbus relations, i.e. the kinds of awareness A and B might have of each other once they are both aware of each other's presence.

4.4.1. Some Graphical Notations

In order to conduct this analysis, we propose some graphical notations for depicting relations between the foci and nimbi of two parties to interaction. Three possible candidates are shown in figure 4.10 below.

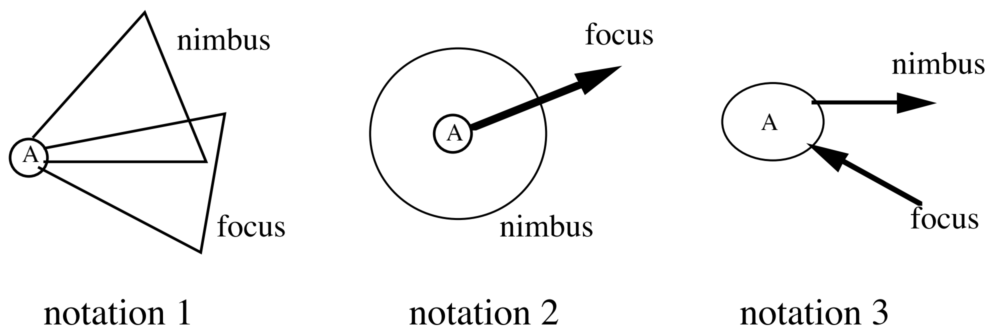
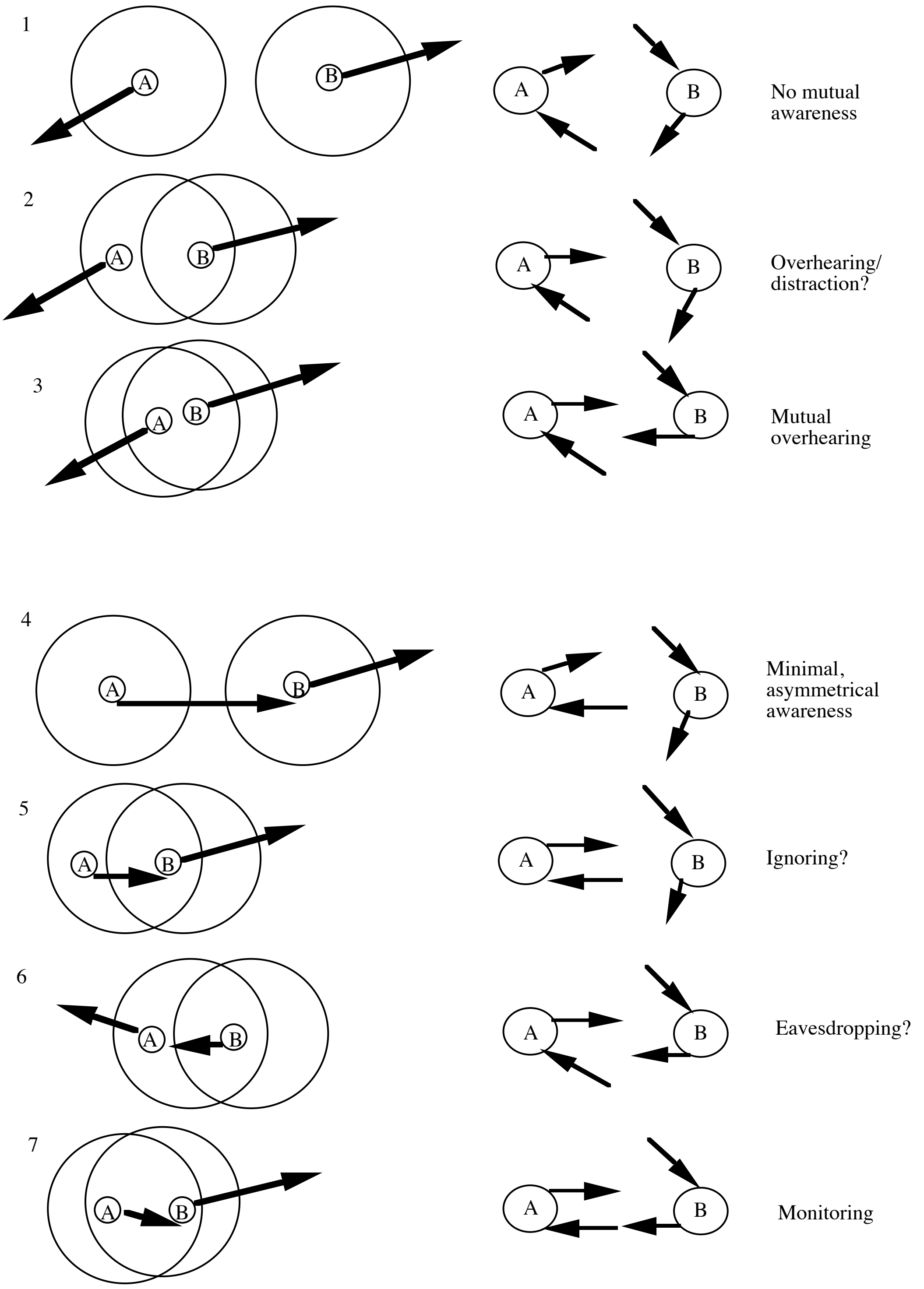


Figure 4.10: Candidate notations for depticting focus and nimbus

Notation 1 has the advantage that a focus of any (2D) shape and size can be depicted (what is shown here as triangles could be circular or whatever). However, focus and nimbus require labelling (or perhaps colouring) to distinguish them fully and pictures become very complex when the inter-relations between even two entities are shown, let alone interactions between a large group. We also propose two other possibilities with a different profile of advantages and disadvantages. In notation 2, an entity's nimbus is shown as a circle, its focus as an arrow. While this enables the distinction of focus from nimbus without further labelling, it is committal over the kinds of shapes focus and nimbus are suggested as being. When a number of entities are shown, this represents a small improvement over notation 1, but relationships between foci and nimbi are still sometimes less than clear. Hence, we propose a third additional possibility, notation 3, which concentrates on the relations between foci and nimbi with a complete loss of 'iconicity' (i.e. the shapes, sizes and absolute directions now have no meaning). In this notation, focus is depicted as the arrow entering the entity, nimbus as the arrow leaving.

4.4.2. Ten Modes of Mutual Awareness

On this simplified model, it is possible to define ten different modes of mutual awareness when we consider a two entity case. We derive ten as follows. An entity may directly project its focus on the other or not. Similarly, it can project its nimbus on the other or not. This gives four possibilities for the modes of awareness one entity can have of another. As we are dealing with a two entity case, we have $4^2=16$ possibilities. However, six of these can be eliminated by symmetry, leaving ten. These are depicted in figure 4.11 using two different graphical notations (notations 2 and 3), together with a tentative name for each mode.


1
A
B
No mutual awareness
2
Overhearing/ distraction?
3
Mutual overhearing
4
Minimal, asymmetrical awareness
5
Ignoring?
6
Eavesdropping?
7
Monitoring

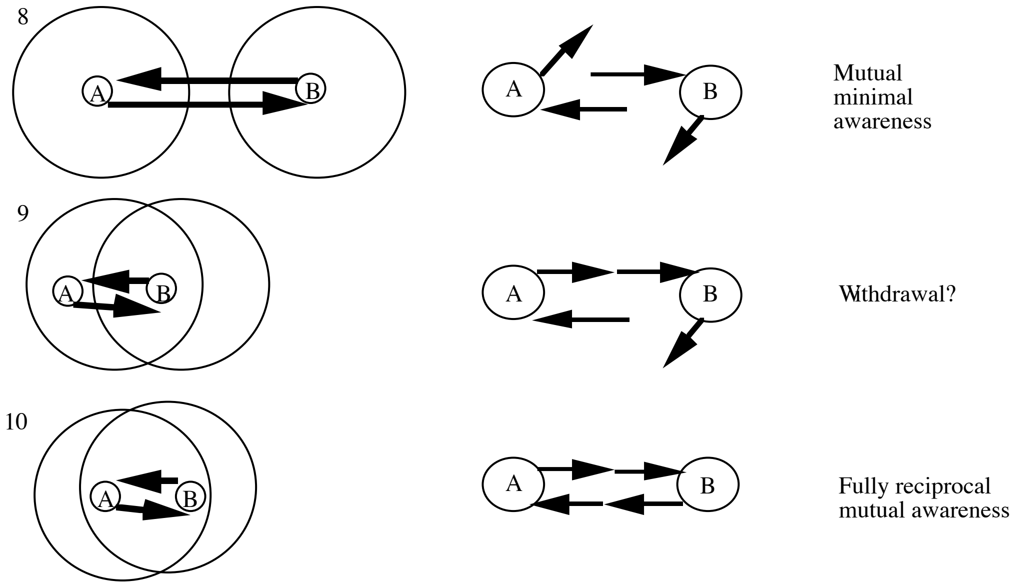


Figure 4.11: ten modes of mutual awareness

Once again, it is important to understand that this represents a rather drastic simplification of the spatial model. In a situation where focus and nimbus are continuous and multi-valued across space (i.e. some kind of field effect), there may be an infinite number of possible levels or models of awareness. However, the following ten would still represent ten key states (i.e. general approximations). It is important however, not to fall into the trap of thinking of the spatial model as actually being defined by these ten modes!

4.4.3. Some Modes of Mutual Awareness Further Interpreted

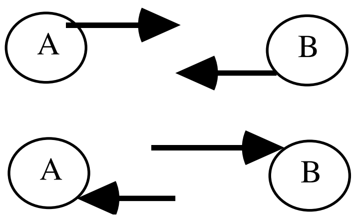
This analysis of the spatial model is of interest because it captures some aspects of previous empirical studies of work practice (see chapter two). For example, case 3, which we have named *mutual overhearing* captures some aspects of the ways in which London Underground control room operators make their activity available *to each other*. Similarly, case 7 involves both parties projecting their activity over each other but here one party is focusing on the other and attending to that party's activity. Again, this might capture some aspects of the mode of mutual awareness when one operator comes to attend to the activities of another. A virtual environment in which participants can move between cases 3 and 7 (and occasionally engage in 10) might simulate many aspects of working spaces where mutual unobtrusive monitoring is important.

Symmetry, Congruence and Reciprocity

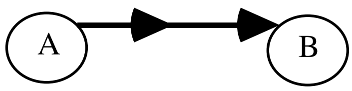
The analysis of these cases suggests certain properties which occur in a number of the cases. This concern relations between the foci/nimbi of the two entities. Let us define *symmetry* as follows: a mode of mutual awareness exhibits symmetry if either (i) A is focusing on B and B is focusing on A or (ii) A is projecting its nimbus on B and B is projecting its nimbus on A. A mode of mutual awareness may be symmetrical in up to two ways — with respect to focus or with respect to nimbus. Now let us define *congruence* as follows: a mode of mutual awareness is congruent if A is focusing on B and B is projecting its nimbus on A. A mode of awareness is *reciprocal* if it is both symmetrical and congruent. It is *fully reciprocal* if it is symmetrical with respect to both nimbus and focus and is congruent.

The third of our graphical notations is especially good at depicting these relations. For example:

Symmetry



Congruence



Transitions

As a final piece of semi-formal analysis of these various modes of mutual awareness, let us consider the ‘one-step’ transitions which can be made from one mode to another. By ‘one-step’, we mean transitions in which there is just one change to just one of A’s foci or nimbi or B’s foci or nimbi. Thus the transition from case 1 to case 2 is one-step but the transition from case 1 to case 3 is two-step. The transitions are depicted in figure 4.12. We also introduce the concepts of *foregrounding* and *backgrounding* to describe transitions between awareness states which either increase or decrease awareness respectively (figure 4.13). To what use can such a diagram be put? First, it is important that such representations be handled with **EXTREME CARE** — they are gross simplifications of the concept of awareness, if only because of the simplifications involved in this (discrete) instantiation of the spatial model. However, perhaps we can point to some tentative possibilities. These named modes and the transitions between them might simplify nimbus/focus manipulations for inhabitants of cyberspace. Rather than saying project nimbus in such a way and direct focus in such a way, they can say ‘ignore’ or ‘monitor’ etc. Secondly, and perhaps more interestingly, one can imagine rooms in which a certain subset of these modes of awareness are possible together with the transitions between them. A virtual control room may be an environment in which the following modes and transitions are the most readily possible. This would be one way of integrating the room metaphor with an interactional approach to space.

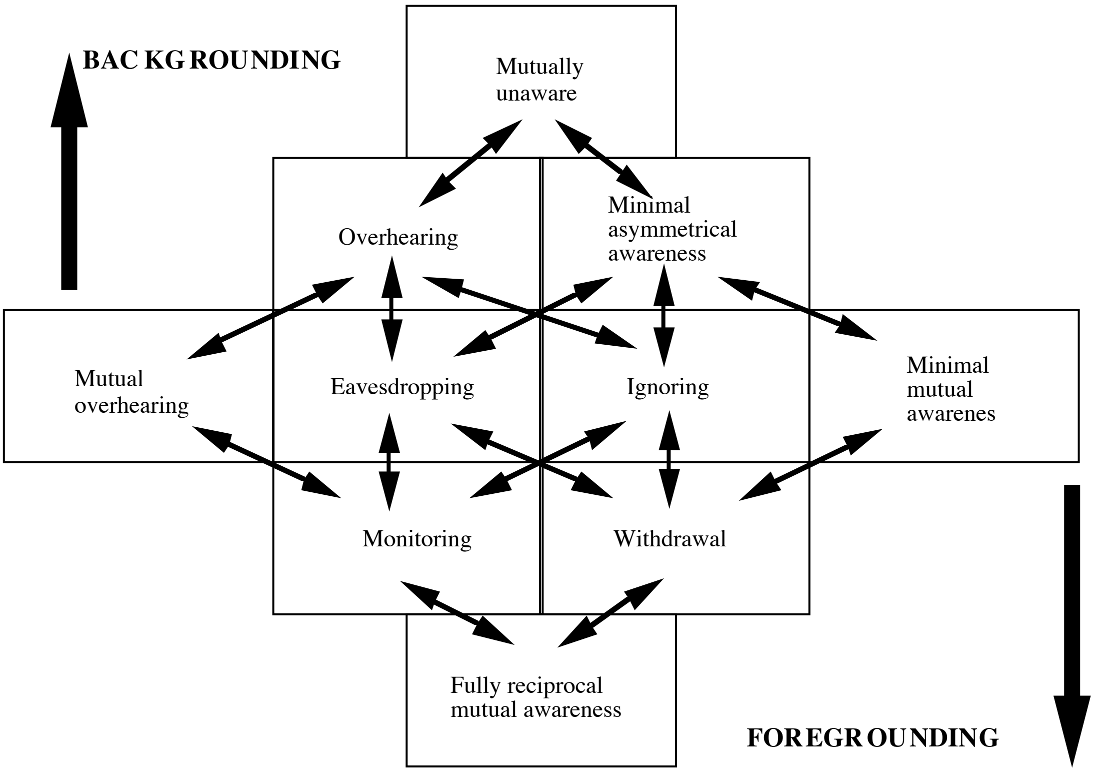


Figure 4.12: transitions between models of mutual awareness

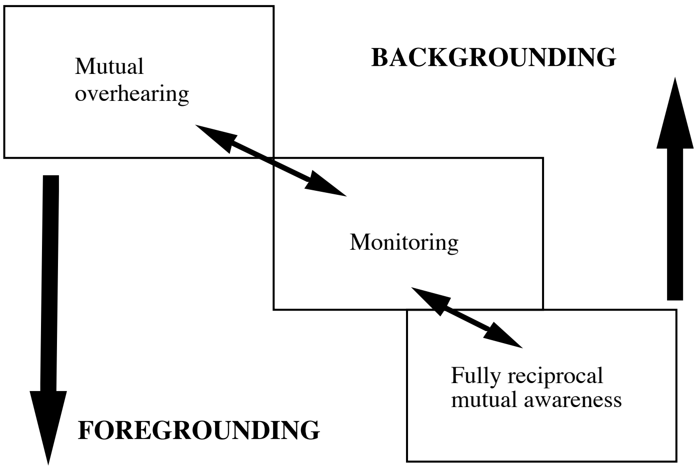


Figure 4.13: Foregrounding and backgrounding

4.5. Distributed Processing Requirements

In contrast to the previous section, we now consider the systems support required for the spatial model from underlying distributed processing platforms. More specifically, we explore the relationship between the COMIC spatial model and the

Trading model for bokering services within ISO's Open Distributed Processing (ODP) framework.

Trading in distributed systems

Much effort has been invested in to the development of platforms for building large scale distributed systems including the Open Distributed Processing framework (ODP) (ISO 1991a); the work of the Object Management Group (OMG) (OMG 1990); OSF's Distributed Computing Environment (DCE) (OSF 1992) and systems such as ISIS (Birman 1991). This chapter assumes the ODP model (the OMG and OSF work is broadly similar; ISIS adopts a contrasting approach).

Within ODP, a distributed system can be modelled as a set of *objects* which use each others *services* via well defined *interfaces*. An interface groups together a set of related *operations* which are invoked by one object on another. A distributed platform provides some mechanism for establishing contact between objects, negotiating the use of services and invoking operations. This is supported through the process of *trading* (ISO 1991b). In order to trade, a provider object exports its interfaces by registering them with a well known system object called the trader. The trader notes the *type* of each interface and also the *context* in which it is provided (effectively the name of the service provided). A consumer object that wishes to use an interface queries the trader, supplying both the desired interface type and also target contexts. The trader looks for a match and, if one exists, returns an *interface reference* to the consumer. This interface reference can then be used by the consumer to invoke operations on the provider. Notice that it is the consumer who decides to request a service and that, in effect, the trader is a passive service. The main advantage of trading is that it provides a high degree of transparency for object interactions. The process of trading is summarised by figure 4.14.

Extensions to the trading model

We expect that collaborative virtual environments will be characterised by a number of features which will impact on the nature of object interactions and on fundamental ideas such as trading. First, they will include objects which represent human beings. Human beings are intelligent and autonomous, often liking to explore their environments. Interaction between objects will therefore often be ad-hoc and opportunistic. Objects will not always know in advance which interfaces they require and so a passive trading model will not be sufficient. Instead, objects will require the trader to actively inform them of new services as they find them (e.g. imagine picking up a tool that you haven't seen before). Second, in addition to interface type and context, trading will be based on the spatial proximity of objects. In other words, as objects get closer to each other, they will become more aware of each other and will able to invoke new operations on each other. In this way the environment becomes more dynamic as objects react to each other's presence (consider the bulletin board application described earlier). Thus, trading becomes an active process based upon the proximity of objects to each other.

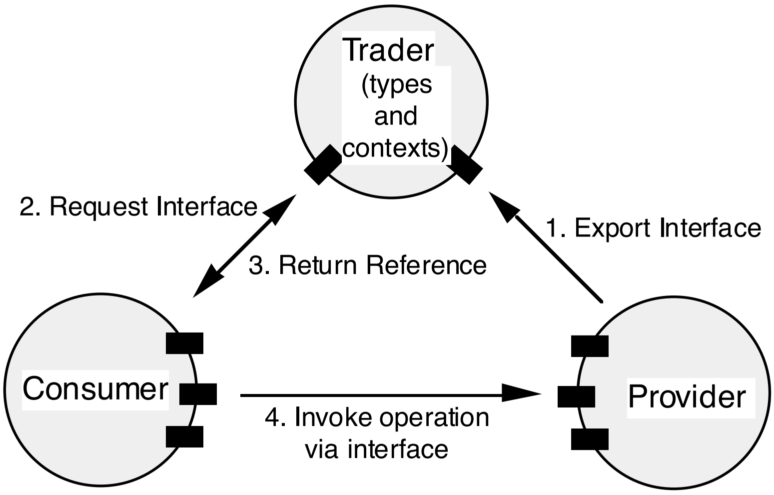


Figure 4.14: Trading in Distributed Systems

We propose to support to notion of active, spatially based trading by integrating the trading model with concepts from our spatial model. At the same time this will show how the spatial model might be realised within an ODP environment. First we consider a general mapping of terms. Objects in the spatial model correspond to objects in the trading model. Communication media are mapped onto different interfaces (e.g. “audio” or “text”) allowing interaction between these objects. Now we can introduce the idea of managing object interactions through inter-object awareness. We can directly associate an aura with each interface. When two auras collide, the relevant interfaces are enabled. In other words, the objects mutually acquire interface references. It is the role of the environment to somehow detect this collision and to pass the references out. Next, we associate focus and nimbus with an interface. This time it is the objects themselves, not their surrounding environment, that negotiate awareness levels. These levels can then be used in two ways. Operations within an interface can be associated with an awareness threshold at which they become available to other objects. Also, objects can decide to invoke operations on others once certain thresholds are passed. This ability for objects to determine levels of mutual awareness requires support from standard operations to return values of focus and nimbus from a given interface. These key extensions of aura, focus and nimbus in object interfaces are shown in figure 4.15.

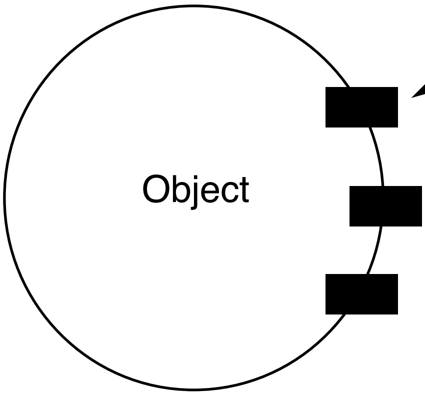


Figure 4.15: Spatial model extensions to object interfaces

Proposal for a generalised architecture

We now propose an outline distributed architecture to implement the extended trading model. First, we consider the general steps involved in object interactions in virtual space. Objects entering and leaving the space need to register their capabilities (i.e. interfaces supported). Aura collisions need to be detected for each interface. More specifically, this process needs to take place whenever two or more objects support a common interface type in the space. Following an aura collision, the objects involved need to be given interface references to each other, thus enabling them to interact. Subsequently, the objects need to use focus and nimbus to negotiate mutual awareness levels so as to control this interaction. We propose partitioning responsibility for these processes across three kinds of object in the distributed environment: the trader which will be responsible for registering interfaces and for passing out interface references; the interacting objects themselves which will be responsible for control of focus, nimbus and awareness; and a set of *aura collision detectors* which will monitor for aura-collisions. In the most general case, there will be a separate aura collision detector for each interface type. Our goal in making this partitioning is to devolve responsibility in as decentralised manner as possible. We now consider the roles of each of these objects in more detail by following through the sequence of events which occurs as two objects enter a space, approach each other and then interact. The five steps involved in this process are summarised in figures 4.16 and 4.17.

1. Object X with interfaces i,j, and k enters the space S. It registers these interfaces with the trader for this space T_S. We assume that one trader will be responsible for each space. Thus, a space corresponds to the notion of a *trading domain*.

2. Object Y with interface i and j enters S and registers with T_S.

3. There are now two objects in S supporting both i and j. Consequently, Ts requests the creation of two aura collision detectors, D_{Si} and D_{Sj}, for interfaces i and j respectively. This creation may involve the use of the ODP *factory* service. T_S passes on interface references for X and Y to D_{Sj} which stores them for later use. T_S also has to pass interface references for D_{Si} and D_{Sj} to the objects X and Y to

support aura collision monitoring. Other objects supporting i and j entering the space at a later time also have to be given references to D_{si} and D_{sj}.

4. X and Y move closer together. Eventually, D_{si} detects an aura collision between X and Y's i interface auras (assume that the aura for i is larger than the aura for j).D_{si} now passes out mutual references for interface i to X and Y. X and Y can now interact via i.

5. X and Y interact. The nature of this interaction is governed via awareness levels (e.g. thresholds for enabling and invoking operations). In turn, awareness is controlled by nimbus and focus which are specific to interface i.

Of course, many issues remain open. First, the proposed architecture has yet to be tested and verified and many refinements may be needed — in essence, the proposed model is a "strawman". Second, it is not at all clear how well the underlying ODP RPC paradigm is suited to this kind of problem. Monitoring for aura collisions across combinations of interfaces may require a more broadcast/multi-cast oriented approach such as that provided by the process group model of ISIS. In particular, the use of hierarchical process groups may represent an alternative approach. On the other hand, ISIS provides little support for registering and brokering interfaces ("entry points" in ISIS terms). It is also interesting to note that ODP implementations such as ANSA (ANSA 1989) are currently moving towards the provision of more group oriented communication models.

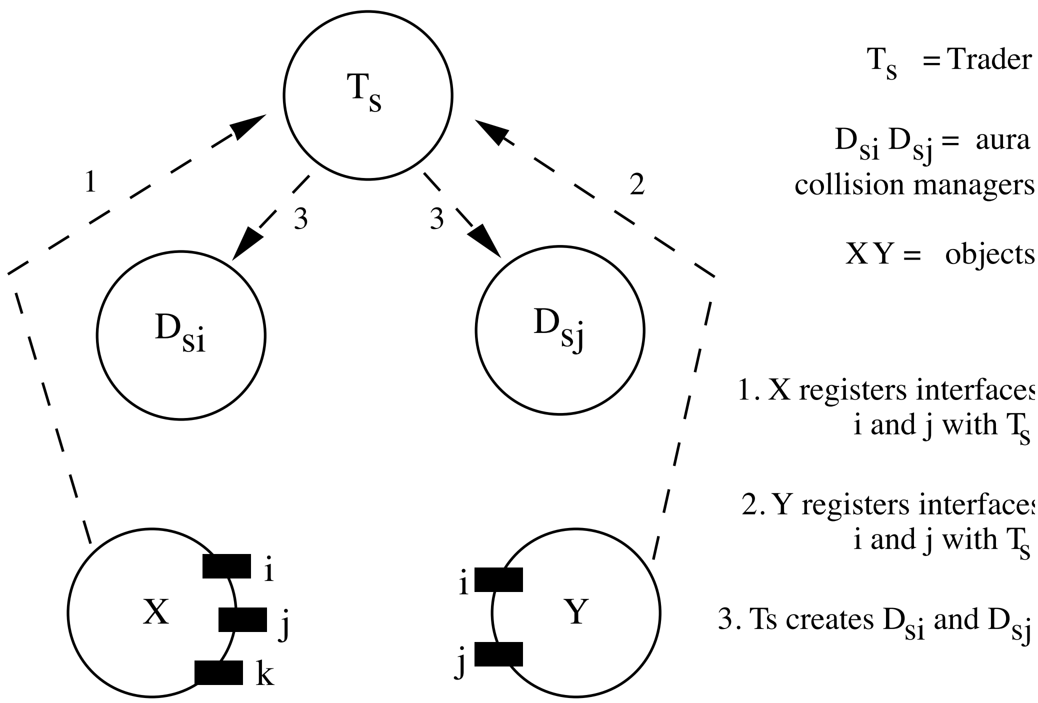


Figure 4.16: Registering interfaces and creating aura collision detectors

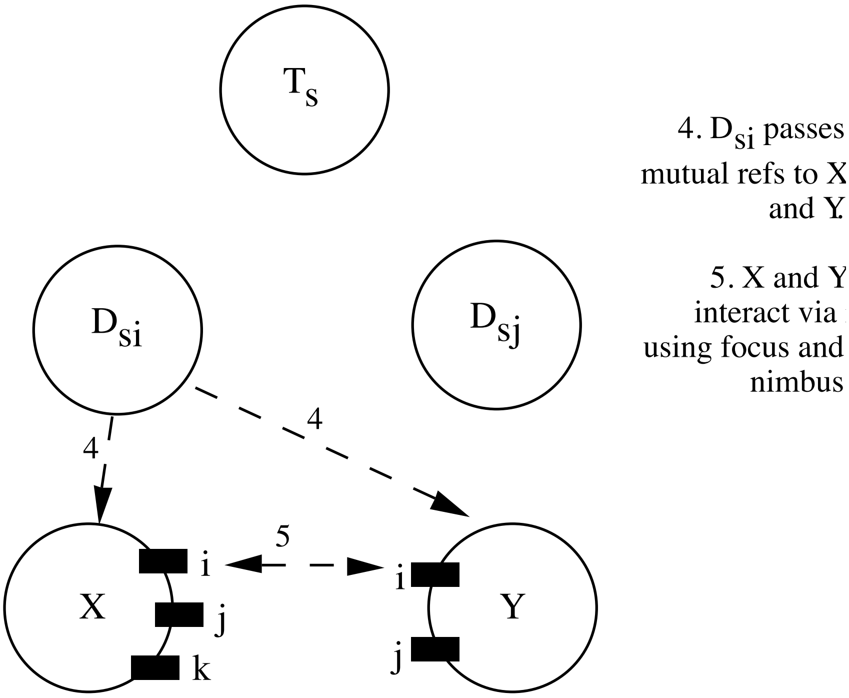


Figure 4.17: Enabling interaction after aura collision

Federated trading across connected universes

We might connect together distinct virtual spaces, perhaps through portals or gateways, to create a widely distributed global universe. In terms of the trading model, this raises the issue of trading services across a distributed set of trading domains. At the very least, this involves the movement of objects between trading domains which, in turn, requires dynamic registration of interfaces and also knowledge of which domains (and hence traders) are logically “connected” to which others. The issue of trading interfaces across domains is known as *federated trading*, and is receiving considerable attention within the wider distributed systems community. This section explores the issue of federated trading within the context of collaborative virtual environments. In particular, it discusses two approaches to the problem of acquiring and managing trading knowledge.

In order to trade across domains, traders must first be aware of each others existence and location. It may also help if they share knowledge of each others current services so as to provide information to client objects about possible domains of interest (i.e. what other virtual worlds are available, how to get to them and what they might find there). We call such knowledge trading knowledge. Emerging approaches towards federated trading involve traders establishing links or agreements whereby they hold knowledge about remote services under some part of the local context space (in other words, cross links or pointers are established between respective context trees).

We now consider two problems. First, once again, it is important to note the dynamic and mobile nature of objects in virtual worlds. This means that trading knowledge, at least in terms of the details of local services, may change rapidly and may not always be accurate or trustworthy. In short, it may not be possible to know which services exist in a given domain until arriving there and only then through a process of exploration. In particular, the idea of establishing links across context trees (indeed, the idea of maintaining context trees at all) may be infeasible. Second, in order to federate, traders must first know of each others existence. How does this happen? (in fact, this bootstrap problem applies to federated trading in general). Also, how does knowledge of new traders spread through a federation?

We consider two approaches to this second problem — relying on a global database of trading knowledge and the use of the mobile client objects or traders themselves to spread the knowledge. The former involves using some kind of global database as a repository of knowledge about traders, their locations and possible services. Examples of such databases might be the X.500 Directory Service (ISO 1988) or the Internet Domain Name Server (Mockapetris 83). However, several researchers have expressed doubts over the political and managerial, if not technical, feasibility of providing such a global service (Van der Linden 1992). The second approach involves the objects within the federation spreading trading knowledge as part of a gradual learning process. One possibility might be for the traders themselves to interrogate each other as to their knowledge of other traders. Thus, a trader could gradually explore the federation, gaining knowledge as it went. A second approach might be for the clients to carry trading knowledge within them as they explore. Registration within a local domain (i.e. entering a new world) might then involve exchanging some knowledge of the outside universe (e.g. the client object could pass on details of the last few domains visited and even some of the services encountered). This client-based approach is analogous to a swarm of bees pollinating flowers in a symbiotic relationship. The general goal of these last two approaches is that, given initial knowledge of one other trader (the bootstrap problem), a trader can learn about many others without the need for either a global database or human intervention. Of course, many issues then arise including problems of preventing a chain reaction of trading knowledge and of dealing with out-dated trading knowledge. However, the general approach seems promising, particularly for large collaborative environments where highly mobile and intelligent client objects are likely.

4.6. Conclusions and some further issues

This chapter has described a spatial model of group interaction in large-scale virtual environments. The model provides mechanisms for managing conversations between people, as well as interactions with other kinds of objects, in spatial settings. The notion of awareness is used as the basis for controlling interaction and the model provides mechanisms for calculating awareness levels from spatial properties of objects (e.g. position and orientation). This allows objects to manage interactions through natural mechanisms such as movement and orientation in space. The model defines the key concepts of aura, focus, nimbus adapter objects and boundaries, all of which contribute to the management of awareness. Furthermore, these concepts are defined in a sufficiently general way so as to apply to any system where a spatial metric can be identified.

The chapter discussed a number of potential and existing applications of the model including its application to virtual reality systems and abstract data spaces (the latter resulting in the notion of *Populated Information Terrains)*. We also presented an analysis of the model in terms of observed awareness phenonema derived from previous field studies of cooperative work. Finally, we briefly explored the relationship bwteen our model and emerging distributed processing models, particularly the notion of Trading within ISO's ODP effort.

Of course many issues remain to be addressed. Using the terminology of chapter two, the current model focuses on interaction at the microscopic level with particular emphasis on support for conversation management and interaction with reactive objects. Among the many issues so far untouched are appropriate embodiment, spatial structures and navigation. However, even within the model's current limited scope, there are several key questions which remain to be answered.

Assemblies: Aura, Focus and Nimbus

The model makes no assumptions about the kind of thing an 'object' is. The Principle of Extended Symmetry from chapter two suggests that objects could represent humans or data. To go a step further, why shouldn't they represent groups or assemblies of individuals? In turn, this would require some way of computing the aura, focus and nimbus of assemblies from the auras, foci and nimbi of their constituents. How might this be done? Put another way, do we need to move the model from its current dyadic orientation towards a more group oriented view?

Computing emergent properties of this sort which can be predicated on assemblies may well also be useful for enabling distinctions between different kinds of participation to be made. For example, as Goffman emphasises (Goffman 81) (see also (Jirotka 1991)), communicative interaction involves much more than a speaker addressing a hearer followed by these roles switching through turn-taking. A participant can have a number of different footings with respect to what is said. Participants can be ratified or not, can be overhearers, can be actually addressed. The participation role of 'hearer' is complex and 'speaker' similarly so. In terms of the concepts we have been working with, we may be able to get at some of these issues by defining the nimbus etc. of assemblies. For example, (i) 'speaking' while focusing on the assembly is a different matter from (ii) speaking while focusing on a particular member when casting one's nimbus over the assembly. In this way, perhaps, different kinds of 'speakings' and 'hearings' (hence footings) can be identified. Perhaps a spatial model which associated groups as well as individuals with focus and nimbus might better support these ideas.

Asynchronous Awareness': Traces and Trails

It may seem from much of the foregoing that synchronous interaction and communication is being privileged over asynchronous forms. Indeed, this impression is fostered by the studies of work practice we have mentioned which concern interaction in time-critical environments (ATC and other forms of traffic control). In virtual cooperative spaces, we conjecture that the *relation between synchronous and asynchronous interaction is the relation between whatever moves and whatever is left behind*. Thus, for example, someone could become 'asynchronously aware' of some event, interaction or of the past presence of some person or assembly, if some 'mark' on/in the space is left behind. "Kilroy was here" inscribed on a wall tells us that someone/thing, possibly called Kilroy but certainly able to use an inscription device like a spray-can was indeed here. Kilroy or her impersonator has moved on, the mark (graffiti) does not. Marks of this sort project aura and nimbus (and maybe can focus too?) much like any entity in the virtual spaces we have been describing.

One can also imagine *trails* which are marks left behind as a result of navigation/traversal through space. Again, a trail might have an aura and nimbus (focus?) of its own which would give information to the hapless traveller in cyberspace (e.g. of who had gone before or where they ended up, the destination 'percolating' back down the trail perhaps!). (Note: maps might 'intelligently' interact with trails!)

Integrating time and space

As its name implies, the spatial model is concerned with the role of space in supporting cooperative work. What of time? Many problems of cooperative work are concerned with synchronisation, scheduling, planning and coordination. One tempting possibility is to consider time as "just another dimension" and to apply our concepts of aura, focus and nimbus to events in time. This may not be as far fetched as it sounds. People could have a temporal focus which determines when in time they are focusing their attention. Such a mechanism could be used to configure alerts and notifications of particular events (e.g. tell me if anything interesting happens between 13:00 and 14:00 but don't bother me between 2:00 and 3:00). Conversely, objects or events might have temporal nimbi to scope their presence in time. Thus, residual awareness of an event might linger sometime after it happened. Another use of temporal nimbus might be to pre-announce your arrival when teleporting to a given location. Thus, people in a group would perceive a growing awareness of your "approach" before you joined them. This might allow us to relax Michael Benedikt's principle of Transit without producing some of the dangerous social consequences he predicts (see chapter two).

Events and filtering

The whole notion of *events* causes us some problems. Our model is concerned with awareness.; but awareness of what? Our current emphasis is on awareness of presence. What about awareness of specific events? For example, If I generate an event does it have a nimbus of its own? Do some events result in greater awareness than others? These questions also relate to the corresponding model of awareness being developed for the COMIC shared object service and to the idea of propagating awareness of changes to objects in a shared object store.

A final question concerns the notion of filtering. Focus supports the spatial filtering of information (“I am interested in these things over here but not those over there”). Humans also employ other kinds of filtering (“I want to be aware of all red objects” or “tell me about all objects whose name contains the word ‘Fred’ “). We need to consider the relationship between our spatial approach and these other approaches.

Clearly, much further work needs to be done. Beyond these immediate issues, the current prototypes require extension and evaluation. Additional applications also need to be modelled and demonstrated. In spite of this, at this stage, we are optimistic that spatial models of interaction such as the one described in this chapter, will form an important aspect of support for CSCW, particularly as new technologies such as virtual reality become more widespread in the next few years.

5. Rules of etiquette at the global cocktail party: issues of metaphor in CSCW design

Mike Twidale

University of Lancaster

The use of spatial metaphors such as rooms and terrains within the COMIC project, and indeed within many others, aims to exploit our natural spatial skills to improve the learnability and usability of the resulting systems. This chapter considers the effects of using metaphors in systems design. It uses the work on metaphor in the literature of HCI, cognitive science and computer-assisted learning to assess the effects of metaphor use in the development and use of CSCW systems. In particular it raises the issue of etiquette in social activity and how this evolves a computer-mediated social setting.

5.1. Advantages of metaphor: A rosy picture

Chapter two explained how the COMIC work on spatial modelling aims to exploit our natural spatial skills, both cognitive and social, to improve the usability of CSCW systems. Chapter three then proposed a spatial model of interaction which employs a number of key metaphors for managing interaction in large virtual spaces (the metaphors of aura, focus, nimbus, boundaries and various adapters such as podia and tables). This chapter refects on the overall effects of employing such metaphors in systems design, pointing to both the advantages and also to the potential dangers and pitfalls of this approach.

There is a substantial literature in the overlapping fields of education research, cognitive science, HCI design and computer-assisted learning on the subject of the use and effect of metaphor and analogy in aiding learning in general and the use of computer applications in particular. For the purposes of this chapter we use metaphor and analogy synonymously; there is considerable debate over the distinction between the two. There are three linked effects of metaphor: on the initial learning of the system, on the continuing use of the system, and for developers on the generation of new applications. The aim of this chapter is to describe the general lessons of the HCI-based approach to metaphor and consider its likely effect on the development and use of CSCW systems.

There are numerous claims (eg. Carroll & Thomas 1980, Rumelhart & Norman 1981) that when designing a computer system to perform a task, a real world metaphor offers a number of interlinked advantages:

- It makes the intent of the application more comprehensible to the user. That is, it can be effective in giving an overview of the application as a whole, which may consist of a variety of sub-applications, all of which will have to be learned. In the absence of such an overview, it is hard for a novice to gain an understanding of the use and usefulness of the interrelationships between the sub-applications. If this problem occurs there is the danger that they will only bother to learn a subset of the sub-applications, and so the real benefits of the application as a whole will not be realised. For a particularly novel application such as may occur for CSCW, it can help to give an idea of the usefulness of the tool and how it can be employed in various forms of work practice. In the absence of such a view, understanding and adoption can be severely impaired (Orlikowski 1992).

- A metaphor as an aid to learning/understanding can offer further benefits by providing 'gratuitous' information. That is, if the analogy between the real-world system and the computer system is close so that there are many additional sub-correspondences, the user may not need to actively learn about certain features because they can infer their use from the metaphor. Likewise they can use the metaphor to effect the mapping from their goals to the available operators to achieve those goals. The metaphor can also serve as a means of error correction. By using the metaphor one of the mappings from the analog to the action structure of the system becomes more meaningful, indicating the correct action.
- If the metaphor used is a near-direct mapping of existing work practice, it can offer advantages during continuing use as well as during the initial learning period. By matching the existing work environment, existing work practices can continue unhampered while benefiting from additional features offered by computerisation. An example would be the desktop metaphor of the Macintosh interface.
- Besides offering advantages, it may be that a metaphor is the only way to initially design a new system. It is very difficult to create a completely new application. Instead one uses existing computer systems or real-world systems as a source of ideas. Also, it is difficult for a user to understand the usefulness of a completely new system (all the other points refer to usability). Thus the best way can be to take an existing system and computerise parts of it. By using a metaphor of an existing successful system or practice one can build on the usability features that have evolved with the original.

5.2. Disadvantages of metaphors: The spectre at the feast

5.2.1. Cognitive Limitations: Every rose has its thorn

A metaphor can lead to certain classic misconceptions by pursuing the analogy too strictly. This is the inevitable consequence of the gratuitous learning you also get from pursuing the analogy. For example, with the typewriter-word processor metaphor this can lead to the use of Return to end lines rather than using word-wrap.

A metaphor can also perpetuate restrictions on activity that had a real basis in the original version but are no longer relevant for a computational version. This can include ignoring the new features of the system that do not map over from the analogical case. An example is the use of the Find command in word processors that is not available on a typewriter.

It can restrict design creativity, because one always thinks only in terms of slavishly adapting existing non-computer systems. As Goldstein (1981) notes of the development of word processors: “If we program these machines to imitate paper ... we will never know if qualitative improvements in the handling of words can be obtained”.

For a complex system, a single metaphor may be insufficient to account for all the activity. This greatly increases the complexity of understanding for the user as she must now decide on which metaphor to employ in each context. There may still remain cases where all the metaphors break down (Halasz & Moran 1982). For example a computer’s file management facilities may be explained by analogy to a filing cabinet. However, this analogy fails for password protection. If one expands the metaphor by describing a filing cabinet with a combination lock, one still has to account for protection of folders and individual files. As more and more special cases are added to the real-world analogue in order to explain the computational system (nested folders, links etc), it becomes increasingly baroque and less like any familiar real-world entity. An alternative is to propose a new analogue for file protection only, such as a guard who retrieves documents only when given the correct password. Now the user is in danger of reasoning about guards when she should be reasoning about filing cabinets opening up endless possibilities for errors and confusion.

Different use of terminology between the system and the analogue can lead to confusion for users. For example in the learning of probability the terms ‘permutation’ and ‘combination’ have very precise meanings. Unfortunately certain real-world analogues that illustrate the concepts use the terms with quite opposite meanings: a combination lock is based on permutations and ‘perm’ in football pools are based on combinations! No teacher would choose to provide students with such dreadful analogies but the danger arises that the students spot the analogies themselves leading to inevitable confusion.

5.2.2. Evolution of metaphors: The growth of dinosaurs

Even though a computer system may be based on an existing system, the process of implementation enables one to add features that are only possible or feasible with computerisation. Some of these features will be provided in the initial version of the system, but many more will be added over the course of time as experience of the system in use builds up and additional requirements are manifest. Note that the provision of the computer tool can also change the nature of the activity it supports so that the requirements are not a fixed set of objective ‘facts’ awaiting discovery but the result of a dynamic co-evolution of the tool and the activity.

As features and options are added to the computer system, the originating metaphor can get increasingly stretched as more features are provided that are meaningless or contradictory in terms of the metaphor. The question arises of whether it remains productive to talk in terms of the metaphor, or do the contradictions outweigh the advantages of employing it for beginners and users. Do old metaphors collapse or simply fade away?

5.2.3. Metaphor: A Pandora's Box of requirements

The Post-it note metaphor has been used in a number of computer systems including the Designers' Notepad. If we look at the originator of the metaphor, we can see why it offers so many advantages: as a form of annotation to documents that is easily removable, can be added to, revised and reattached in a different location. It is interesting to note how ubiquitous the use of Post-it notes are: not just attached to paper documents but also to doors, tables, books, walls and computer screens (particularly the border). The implication is that the power of this form of annotation should be provided in any computerised version. That is, the same set of desirable features should be available. There may be additional features possible in the computerised version, but if any of the real-world features are missing, users are likely to find the metaphor irritating as it promises more than it can deliver. Thus the use of a metaphor can lead to the generation of expectations, needs or requirements that the user was hitherto unaware of.

5.3. Is there a way of analysing metaphors?

Can we aim to build up something of a science of metaphor analysis so that after brainstorming a number of crazy metaphors, we can subject them to some form of systematic analysis to clarify their potential usefulness and flaws, to indicate productive ways in which they might be extended and refined to maximise utility and minimise disadvantages?

At this stage probably not, but we can begin the work by setting up case studies of how metaphors, have been used and attempt to find rules for classifying successful and unsuccessful metaphors.

By analogy(!) with HCI research, we can feasibly hope not to create criteria for spotting world-beating ideas, but perhaps rules for avoiding real lemons. That is, most HCI advice is negative rather than positively prescriptive (eg DON'T have too many colours, DON'T require the user to memorise a lot etc) Similarly we can aim to produce some negative rules for metaphor construction. For example:

- Rule 1: DON'T pick a metaphor that your target user population may not be familiar with.

(Not as obvious as it sounds: metaphors derive from social-cultural experience and what may be common knowledge and experience in your group may not be much use outside. The Academic-Conference metaphor would not be much help to a secretary.

5.3.1. Metaphor case studies

Bridges

When Abraham Darby built the first cast-iron bridge at Coalbrookdale in 1779 his design imitated the pattern of the then more familiar stone arch rather than exploiting the properties of the new material (Lawson 1980).

Tractors

One of the earliest motorised tractors was controlled by reins (Gentner & Grudin 1990). This seemed a sensible control device, being familiar to farmers who were used to controlling horses. It also offered a smooth transition path up to the new technology, because it was so similar to the control strategy used for the equivalent task using the old technology. It was of course a failure and was superseded by the steering wheel. Why? Partly because the nature of the activity changed with the new technology. You could drive the tractor faster and so better control was needed. Also better control was possible by use of a steering wheel.

Spreadsheets

These were directly based on an existing paper-based tool used in financial calculations. The paper spreadsheet was successful and better than conventional ruled paper for its purpose. Computerisation permitted the addition of powerful new features such as programmed computation, templates and recalculation. Over the years many additional features have been provided. As a computer tool the spreadsheet has become widely popular, far more than the paper version was. This leads to the interesting consequence (violating Rule 1) that the vast majority of new users of spreadsheets have never come across its motivating analogue, and yet the spreadsheet as a concept seems relatively straightforward to understand. It is interesting to consider why this is the case. I suspect it is to do with the visibility of the effects of change on the spreadsheet.

The spreadsheet is an example of where metaphor is used to inform the development of the system, but does not have a direct effect on ease of use. The fact that novices do seem to find it easy to learn the general concepts of spreadsheet use illustrates that accessible metaphors are not essential for the successful learning of a system. However, by being based on a successful analogue, it inherits (and substantially adds to) the tested ease of use of the latter.

5.4. Metaphor in CSCW

A number of metaphors have been employed by CSCW developers. These include big sheets of paper (Wolf et al. 1992), whiteboards (Whittaker et al. 1991), glass screens (Ishii & Arita 1991), Balinese Shadow Puppets (Tang & Minneman 1991), the postal system, telephones, rooms (Lauwers & Lantz 1990) and windows (Fish et al. 1990, Root 1988). The one I want to discuss in detail to illustrate some general points is a metaphor which I shall call the ‘global cocktail party’ which seems to be growing in popularity and has implications for the work done at Nottingham and SICS.

The essential features of a real world cocktail party relevant to this use of the metaphor are that it consists of a significant number (more than 10) of participants co-located. Individuals have their own personal goals which require conversation with a number of other individuals. They achieve this by forming sub-groups of two or more for interaction. These groups grow and shrink as people choose to move to other groups in pursuit of their goals. In addition to any immediate goals, the individuals may obtain useful serendipitous information that can lead to the setting of further goals.

As if all this wasn't complicated enough, there is a time dimension. Not only may the party have a definite (but usually flexible) start and finish, but individuals may leave and arrive at different times. Furthermore there may be a host/hostess who considers one of her goals to be responsible for maintaining the dynamics of the party, introducing individuals who are expected to appreciate this and perhaps encouraging the breaking and reforming of groups.

Given the goal to talk to various people and perhaps search for unspecified information, a participant must not only monitor the activity of her group, but also that of other groups in case an opportunity to talk to a particular individual arises. This monitoring can be visual, which has the advantage of not interfering too much with an existing verbal interaction, but it can also be auditory. One of the remarkable abilities of humans is the ability to selectively attend to one conversation out of many going on simultaneously. Auditory monitoring is very valuable in providing information about the content of another group's interaction, at the cost of losing information about one's present group.

This brings us to an important issues for all forms of social interaction and hence necessarily for CSCW, that of rules of etiquette. In the case of the (real) cocktail party this is complex, revealing numerous issues. At the party, conversations are not as private as normal, people are allowed to join groups, and perhaps to overhear or tune into groups as a prelude to choosing whether to join in. However, visible monitoring of other groups while participating in a group is considered bad manners as it implies minimal commitment to the group. Nevertheless monitoring does occur but in a subtle manner.

5.5. Using the cocktail party metaphor in CSCW

A major social problem arises because of computerisation, that the metaphor is no longer bound by space. That is, we can have a global cocktail party with hundreds of thousands of guests. Technically there may eventually be nothing to stop this (the Internet population is of this order already) but socially huge problems remain for the user to manage all these subtle real-time interactions. It is simpler to see this by considering the other interaction metaphors we use.

For electronic mail, the metaphor is naturally the paper mail (but also has aspects of telephony: see later). This is asynchronous and democratic: anyone may write to anyone else. Mail can generally be read faster than it is created and it is possible to filter it (with a human secretary for paper mail, or monitor programs for email).

Until recently there have been few problems with telephones either. Telephonic communication is exclusive. However, new services offer the ability to indicate to a phone user that another person wishes to speak to him. This has been advertised by British Telecom with scenarios of inessential conversations blocking the line and the advantages of being able to interrupt with an important message. The extent to which the consequences of this feature have not been thoroughly explored is illustrated by considering the potential irritation of circumstances where the situation is reversed: an important conversation gets cut short for the benefit of a (more) trivial one.

Consider the problems in cscw-cocktail-party mode. There may be restrictions on entry (admission by invitation only). But within the party, anyone can approach you and engage you in interactions. At the same time the people you particularly wish to see are also in danger of being mobbed. You can't prioritise or make appointments in this metaphor, and so whether you achieve your interaction aims can be semi-random. There need to be ways of coping with the problems of large numbers of people and excessive popularity. In the real world analogue, this is done by techniques such as exclusivity (only a select group of invitees), hiding, disguise, minders or simply avoidance. It is possible to envisage computational analogues for all of these.

5.6. Metaphor and etiquette

A whole new etiquette of interrupting, cutting short conversations and including and excluding participants will need to be developed. In the real cocktail party these things can be awkward but are made easier by the high communication bandwidth for interaction using body language, facial expression, etc which can be used both with participants in small subgroups and for long distance communications for monitoring and establishing future subgroups.

5.6.1. Mixing metaphors: a lethal cocktail

We can expect etiquette problems to arise in CSCW systems whatever the metaphor used. For example, Cool et al. (1992) describe how the use of Cruiser's hallway metaphor leads to the development of a new etiquette, including rules for the interruptibility of Cruiser conversations. In particular they note how with the initial introduction of the system the social norms evolve very rapidly. A contributing factor to the complexity of developing this etiquette is that although based on a metaphor of a hallway and walking into people's rooms, Cruiser is also analogous to telephoning which has a quite different etiquette, particularly with regard to interrupting other people's work and including third parties in the conversation.

Electronic mail also can be regarded as the product of a mixed metaphor. It has some features analogous to paper documents (normal mail) and others more analogous to telephonic communication. This ambiguity has contributed to the debate in the USA on the legal privacy status of email (Elmer-Dewitt 1993). A US government judge has barred the Bush administration from erasing tapes of email on the grounds that they were documents like paper. Nevertheless in general private use, the assumption remains that private message such as gossip should carry the same legal protection from third party monitoring as telephone conversations. However the Electronic Communications Privacy Act of 1986 only prohibits “outside” interception of email by a third party without proper authorisation. Within the organisation which owns and uses the system it is quite legal for supervisors to monitor employees’ conversations.

5.7. Computerising can change the rules of the game

Robinson (1993) illustrates the concept of common artifacts with the example of the hotel keyrack. In addition to its main purpose, it serves a number of other useful purposes. One of these is that with a publicly visible keyrack, it is possible for anyone standing at reception to tell at a glance whether a resident had picked up their key (and so was in the hotel) or not. One of the points of the argument was that if we fail to discover the additional uses of a common artifact, then an attempt to computerise the system will potentially result in a loss of these features. Thus computerising the keyrack may prevent the monitoring of the presence in the hotel of the desk, unless extra features to permit something like that were added to the program. Even then the option may still not be used if it requires extra effort to access. The advantage of the real-world keyrack is that the extra information is provided for free. The problem is that we seem to have a different etiquette for what is acceptable in a physical system compared to a computer system, and are generally more strict about computer systems. So in the hotel keyrack case, even if the computerised system had a facility to query if someone was in their room, it might be considered unacceptable to offer such a facility because it could be abused by for example tracking the position of single female residents. So a computer feature that offers identical functionality to a physical feature may be scrutinised more carefully and considered unacceptable precisely because it is computerised.

5.7.1. Further examples of acceptability change

Now, we breifly consider a few further examples. The Xerox Active Badge, when first described leads to widespread feelings of disquiet at the loss of privacy. However, we may consider it a computerised extension of the open-plan-office metaphor where mutual visibility may be quite acceptable.

People are inclined to mistrust the inclusion of sensitive data on computer databases when they will countenance its use in the form of card index files (the real world metaphor of the database). At times this difference in acceptability may be due to the difference in nature of the two formats (computer data can be duplicated much more easily and analysed quicker and in many different ways) but sometimes it does just seem to be an issue of the form of the data (when despite assurances about the methods of use, dissemination and persistence of the data, computer formats are still unacceptable but paper formats are acceptable). Indeed this difference in social acceptability is enshrined in legislation: according to the Data

Protection Act (1984), the recording of all personal data in digital form requires notification to a Data Protection Officer. For the purposes of the Act, personal details can be as little as a person's name and address. By comparison, there are no restrictions at all on the recording of data onto paper. In some countries (such as Sweden) this manifest anomaly has led to the introduction of laws of privacy that apply regardless of the recording medium. What is significant here is that it is the act of computerisation that changes the etiquette of acceptability that is then applied back to existing systems, rendering them now unacceptable.

5.8. Metaphor: Wrapping it up

Metaphors offer substantial advantages as inspiration for developers of practical and usable systems. They also help the user to learn and to continue to use the system. There are though problems that arise with the use of metaphors that developers should be aware of.

Even though a metaphor may seem to be useful for a CSCW application, the application that embodies it may be unacceptable or unusable because it violates certain rules of etiquette. These rules for the computer system may not be the same as those that apply to the real world system. Just as these rules of etiquette can arise, so they can be revised with familiarity. However the system would need to offer substantial benefits for the user for her to overcome her reluctance to break existing etiquette. The continuing incremental development of a CSCW system participates in a co-evolution of user requirements and etiquette.

6. Assessments of the Room Metaphor: An Interface Model for the Collaborative Desktop

Hans Marmolin

KTH

Following on from the previous chapter on the general advantages and dangers of metaphor in CSCW systems design, this chapter examines the 2D room metaphor used in KTH's Collaborative Desktop system. It then undertakes a theoretical assessment of the usefulness of this metaphor as an interface model by comparing collaborative aspects of rooms in the real world with the current version of the Collaborative Desktop.

6.1. Introduction

This chapter continues our discussion of the use of spatial metaphors in the design of CSCW systems. Whereas chapter four discussed the general merits and dangers of this approach, this chapter analyses the application of a specific spatial metaphor, that of virtual rooms, to the design of a particular CSCW system, KTH's Knowledge Net. Chapter one described how the room metaphor has been used by many researchers to model different aspects of private and collaborative work spaces. Some of these researchers, use the room metaphor as a guiding vision, while others use it as an interface metaphor. This chapter is concerned with the use of the room metaphor for modelling interfaces.

Marmolin et al (1993) have explored mechanisms for knowledge sharing and integration in a distributed environment for collaboration. They propose an infrastructure, the KnowledgeNet, modelled as a distributed "library" of documented and undocumented knowledge. The library consists of expertise declarations, describing the kind of knowledge the member possesses and of the documents produced by the members. Both expertise declarations and documents have links to the originator of the information in a way that supports communication and makes the information space peopled. Sharing and integration of knowledge is then obtained by using appropriate CSCW tools. They also propose an interface to this environment, CoDesk, which is a a generic, tailorable and object oriented collaborative desktop focused on the management of collaboration instead of files. They use a 2D room metaphor to model this interface as a set of virtual rooms or spaces within which people interact and claim that the management of a collaborative environment could be described as the management of rooms in an organisational environment.

The purpose of the chapter is to make a theoretical assessment of this interface model (*the target domain)* by comparing it with collaborative aspects of rooms in the real world (*the source domain)*. Assessments of metaphors can be done both empirically and theoretically. Empirical studies focused on studying end users ability to understand and handle CoDesk are in progress. Both laboratory studies with interface prototypes without any real functionality and field studies with working prototypes have been performed (Ahlström et al 1993). This chapter aims at a theoretical assessment addressing the following questions.

Which aspects of CoDesk are not consistent with the source domain, i.e. the use of rooms in collaboration in the real world? Such aspects could result in users meeting features they not expect when making a metaphorical mapping.

Which aspects of the use of rooms in collaboration in the real world are not realized in the target domain, CoDesk? Such aspects could result in that the users expects features that do not exist.

Which aspects of CoDesk are consistent with the room metaphor, i.e. will result in a valid metaphorical mapping? Then the user could apply earlier experiences of the use of rooms in the real world to understand and handle CoDesk.

Before answering these questions, the source domain (i.e. collaborative aspects of real-world rooms) and target domain (i.e. interface model for CoDesk) have first to be desctribed.

6.2. Collaborative aspects of rooms in real world

What is a room ? A room is something more than a space. It is a space with boundaries. But more importantly it could be described as a space devoted to some activity by social conventions or as a space that affords some activity (Moran & Andersson 1990, Gaver et al 1992) or has some purpose. For example a kitchen is devoted to cooking and furnished to support the activities involved in cooking.

As chapter twos and three explained, the social significance of rooms is evident both in the work of social theorists and in ethnographic studies of collaboration in work (Hughes & King 1993, Heath & Luff 1991). Rooms provide a spatial organisation for social interaction. They define "boundaries in space affording different modes of participation alongside different modes of awareness". At the same time, rooms provide awareness of on-going activities and afford "seeing at a glance".

A room is persistent, i.e. it exists independent of its users. However it is not a static object, but adapted to the needs of its users. Some rooms are permanently owned by their users. They are private offices. Others, as conference rooms, are taken-over temporarily by its occupants through an act of social negotiation. Still others are public as hallways and libraries. Rooms can also be re-arranged for different purposes. A private room can be transformed to a conference room, a room devoted to meeting to a storage room.

Chapter two also pointed out two other important features of rooms. They are both spaces for interaction and spaces and means for assembling related data, i.e. there are both informational and interactional aspects of rooms. Rooms could both be considered as containers of persons and tools for getting the job done, and as spaces for collaboration between members of a team. It has, for example, been shown that in the real world geographical distance between rooms is one of the most important factors for informal meetings (Kraut et al 1986). However, the spatial model proposed by chapter three uses the room metaphor as a source for defining important design concepts irrespective of their usability as interface metaphors. In contrast, the 2D room metaphor in CoDesk is an *interface metaphor* aiming at representing social concepts.

To summarize, rooms function as a kind of artifact in accordance with Robinson's (1993) discussion of the importance of artifacts for cooperative work. Rooms are multifunctional tools for getting the job done, for supporting peripheral awareness, interaction and implicit communication. Rooms provide organisational awareness, awareness about their owner and communicational awareness. Rooms constitute an organizational map for navigation in an organization, for locating people. Rooms provide information about the status and activity of the member by means of open doors, whiteboards with "in and out" information, busy lamps etc. Rooms are also communication nodes. One calls a room, or person in a room. One can leave documents in a room and one can put messages on the door of a room and one can visit a person in a room for information exchange.

6.3. The interface model in CoDesk

In CoDesk, the room metaphor is mainly used as a basis for designing an integrated, tailorable and tool oriented collaborative work environment, a collaborative desktop. This metaphor provides a direct manipulative way of accessing tools, forming groups and structuring the collaborative environment. It provides social awareness and supports informal, spontaneous meetings by decreasing the perceived distance between the members of the KnowledgeNet and by inviting members to communicate when they "use" the same room. It provides social awareness by electronic hallways where one can observe the office of the member and get information about that member's status and activity. Video information about activities in real common rooms as hallways and coffee rooms is also used for that purpose.

A set of simple interface prototypes or mockups have been developed of CoDesk to define the interface model (Marmolin et al 1991b and Sundblad et al 1992) and a prototype with restricted functionality but based on a general object oriented program architecture and a distributed database has been implemented (Marmolin et al 1992) to allow user studies.

The interface model in the present version of CoDesk[1] models the KnowledgeNet as a building consisting of a collection of hierarchically structured rooms representing both public, shared and private spaces containing members, tools and documents.

---

1 The interface model described is based on the last interface prototype. The working prototype of Codesk has not yet implemented all features of this model.

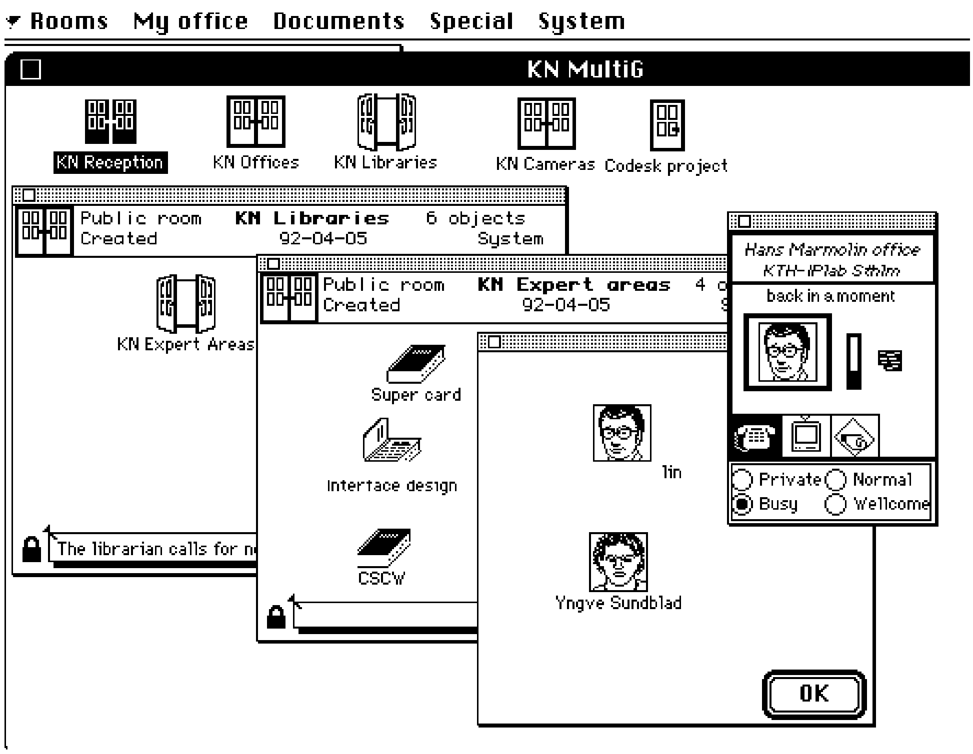


Figure 6.1. An early interface prototype of CoDesk

6.3.1. Public rooms

Public rooms represent information generated by the KnowledgeNet system, that is shared by all members. There is a public reception with catalogues, a public library with documents and expertise areas linked to the originators of the information as described above and public rooms representing virtual hallways and real public spaces as coffee rooms. The use of some public rooms is shown in Figure 6.1.

The Reception

The reception contains catalogues with information about the KnowledgeNet. One catalogue contains information about the members. When the catalogue is opened a set of icons representing the members of the KnowledgeNet is shown. These icons are used to call members by CSCW tools and to get more information about the members. Copies of the icons can be be moved into private rooms.

Another catalogue contains the CSCW tools that the members work station is equipped with. It contains icons representing all the tools that are available for the member. These tools can be accessed directly or copies can be moved into other private rooms. As an invitations to communicate, some of these tools, display members using the tool at the same time. For example the bulletin board displays members meeting at the board, a printer tool can display members lining up at the printer etc.

In addition there are catalogues with information about projects, teams, frame of references, who-knows-what etc.

The Office

The hallway contains a set of offices, one for each member. The user can place offices on the desktop in a way that maps a real or virtual organisation. These offices display information about the status and activity of the members in the same way as open doors, busy lamps etc. give social awareness in a traditional environment. Offices provide support for the owner to set his/her status and for the visitors (the other members) to invoke functions that informs about changes in status. In addition, these offices show the CSCW tools that the owner has access to and if they are used by the owner. These tools can also be used to call the owner. Finally, the owner can display short messages to the visitors, as “out for lunch” and the visitors can leave short messages to the owner, such as “call Hans” or a document.

The Library

There are two libraries, one with documents produced by the members and one with expertise areas. Document can be borrowed or subscribed. The libraries have “a show links”-function and shows the author of a document or the expert responsible for an expert area and how it is defined.

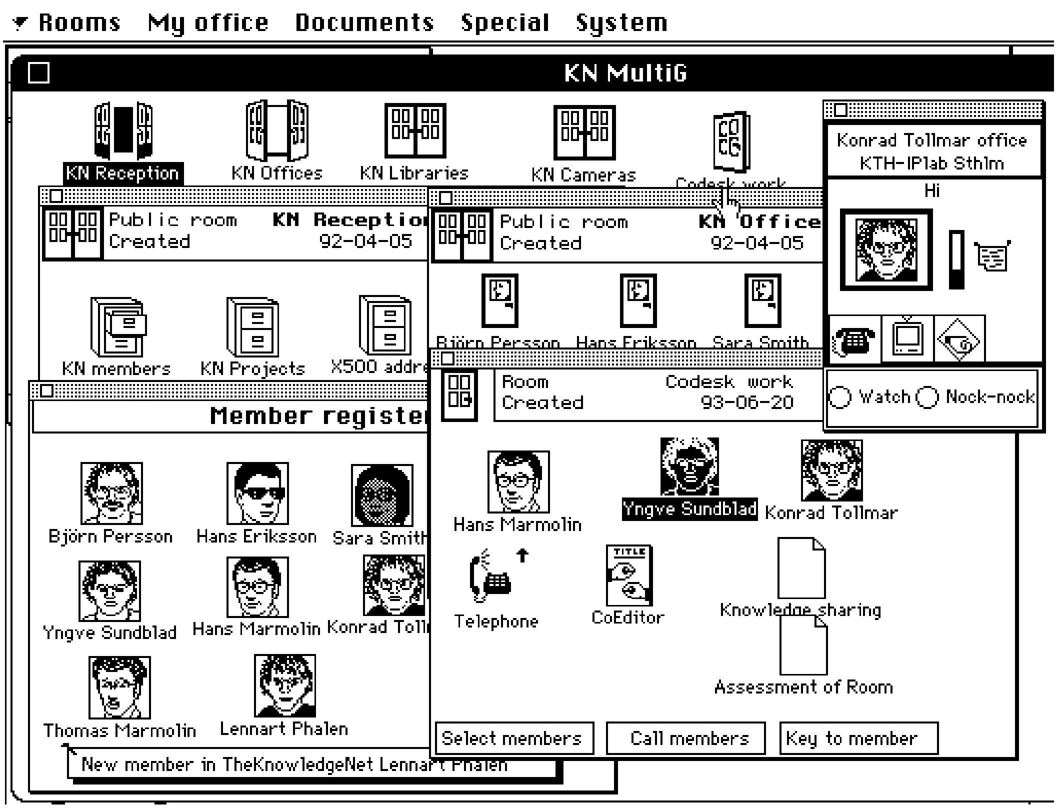


Figure 6.2. The creation of the private room “CoDesk work”

6.3.2. Private rooms

Private rooms are containers for members, tools and documents representing the collaborative work context for each member. They are used as Card & Henderson's rooms (1987) to structure the collaborative work environment in a task oriented way. Private rooms are created and furnished by the members themselves and only displayed at the member's own desktop. The use of a private rooms is shown in Figure 6.2.

One important feature of private and shared rooms is a procedure for *making calls.*. The owner of a private room and the members of a shared room can press a button that recreates the collaborative work situation defined by the room. That is, the tools in the room are adapted to the context defined by the room with the members as default addresses.

Another important feature of private rooms is that the can be shared by the members in the room, i.e. the other members are given a key to the room. It should be noted that a shared room is not a new kind of object, but "shared" is an attribute of a private room. When a private room is made shared it is displayed at the desktop of those members that have been given access rights to the room (have the key) and all objects in the room are shared by these members. For example they have access to the same documents and some instances of CSCW tools, as the bulletin board, are accessible only by the members. Shared rooms represents more persistent subgroups and a shared context for collaboration. They are completely user controlled and there are no formal restrictions related to access rights, floor control etc. All the members that share a room has full access rights to the objects in the room and can delete and modify objects, can furniture the room with new object etc. To support social awareness, a shared room shows which of the members are using the room (have opened it).

6.3.3. Basic objects

The rooms are used to structure the basic objects in CoDesk. The basic objects are members, tools and documents.

Members

Members are the users that have joined a KnowledgeNet. The are displayed as face icons. .

Tools

Tools are any application installed into CoDesk, i.e. could be started from CoDesk. They are represented as pictorial icons.

Documents

Documents are multimedia information created by a tool in CoDesk or copied into CoDesk. Documents could be private or shared.

Expertise Area

An expertise area is any domain of knowledge that a member of the KnoweldgeNet has declared as being knowledgeable in. Expertise areas are represented as pictorial icons with links to the experts

6.4. Assessment of the interface model in CoDesk

Caroll et al (1988) have proposed a method for theoretical assessment, in which one studies how the target domain and source domain maps in a set of key scenarios with respect to tasks, procedures and appearance. This method is here used to assess the 2D room metaphor in CoDesk.

Another more object oriented method is also applied. The interface model in CoDesk will be described in terms of objects, their attributes and methods and judgements will be made of in what degree users could expect these properties and methods, when making a metaphorical mapping. Evaluation is made both on a conceptual level and a more concrete level.

Of course, these kind of assessments assume that the description of the source domain, the use of rooms in collaboration, is valid. Then any differences between the source and target domain, between the structure and relations of objects and their attributes and methods in the real world and the interface model used in CoDesk are potential sources of misunderstandings and confusions.

6.4.1. General differences between source and target domain

There are three inevitable discrepancies between the source domain and the target domain, the first one on a conceptual level, the other on a more concrete presentation level.

Firstly, in the source domain there is both direct and indirect interaction, i.e. both person to person interaction and interaction via some tool as a telephone. In the target domain only the latter is of course possible

Secondly, in the source domain the users point of view is the basis of their experiences. They perceive an inside-out representation of the world. They never see themselves. In the target domain, an outside-in representation is applied. That is, the users have a bird view of the world. They see representation of themselves as well as of others. Although the inside-out representation may make the metaphorical mapping easier, the outside-in representation is necessary in order to be able to manage the role of oneself in collaboration.

Thirdly, the interface model used in CoDesk is 2D, while the source domain of course is 3D. A 3D representation could be used as in Fahlén's virtual reality (Fahlén 1991). However, although the metaphorical mapping may be more simple, a concrete 3D metaphor offers a new set of interface problems.

6.4.2. Assessments of properties and behaviour

The interface model use in CoDesk could be described in terms of objects, relations between objects, their properties and behaviour. By structuring the interface model in this way and estimating the users' expectations of object relations, object properties and behaviour as compared to the description of the source domain one kind of assessment can be made.

Objects and object relations

CoDesk consists of Rooms, CSCW tools, members of the KnowledgeNet, expertise areas and documents. Rooms are devoted to different activities. They are divided into private rooms and public rooms. Public rooms consist of a hallway with offices, a library and a reception. The library consist of a document library and a library with expertise areas. The reception contains catalogues with member, project and tool information. There also are public rooms representing common areas as coffee rooms. Offices can be structured to map the users collaborative environment, for example in terms of real or virtual organisations. Private rooms are member rooms devoted to some specific activity. Members can access each others private room if they have been given a key, i.e. rooms can be shared. CSCW tools consists of synchronous and asynchronous tools, named as corresponding tools in the source domain, e.g telephone, mail, bulletin board etc.

This object-structure corresponds in general with the object structure of the source domain. There are two main discrepancies, related to expertise areas and conference rooms.

Expertise areas are a new kind of objects not expected from knowledge about the source domain. However, they are a fundamental concept in the KnowledgeNet although they do not map to any familiar object in the source domain.

Conference rooms could be expected from the source domain, but do not exist in the target domain. A lot of effort has been devoted to solve this problem, but we have not found a satisfactory metaphorical mapping. In an earlier version a shared room was a separate object that represented a kind of conference room. However, then the basic discrepancies related to direct and indirect interaction and inside-out and outside-in representations become very obvious. For example, does a member icon represent the member or the member's address? Is a telephone in a conference room used to call people outside the room or people inside the room? In addition, the meaning of these shared rooms or conference rooms is confusing. Fore example, in the target domain, shared rooms are used to represent persistent collaborative groups, while in the source domain conference rooms are used for temporary meetings between members in a group. After having analysed this problem in detail, we concluded that having a separate object called "shared room" resulted in inconsistencies in the interface model and in the metaphorical mapping. However, to meet the need of being able to share a collaborative context, the solution described above was chosen. Private rooms were giving the attribute "to be shared", represented as giving members keys to the rooms. That is, a private room that is shared represents a private way of structuring the collaborative environment in terms of members, tools and documents, that other members may use.

Closed rooms are represented as door icons and open rooms are ordinary windows, tools as pictorial icons and members as face icons. The discrepancies on this more concrete level concerns the representation of members and rooms. The member icons really represents the members' addresses and not other features of the members. However this is a nice way of creating a peopled information space and should not be very confusing. Open rooms are represented as ordinary windows. The reasons for not using components that looks like rooms are mainly technical, but in this way CoDesk is consistent with other kinds of desktops.

Public Rooms, Documents and Expertise Areas

Public rooms in CoDesk are mainly containers, although some of them also can be used as interactional spaces. For example, as an invitation to communicate, users of a public room can be made aware of other users using the room at the same time by showing their icons. Public rooms are also used for communication between system (administrator) and members. For example system messages can be shown in public rooms. The behaviour and properties of public rooms and their content are mainly controlled by the system, but the users can open a public room by clicking on the door and close it by using a close button in the window and they can copy objects from the reception, such as members and tools, and documents from the library to other rooms. In addition, when they click on a document or expertise area the icon of the corresponding author is shown and can be used for calling.

The properties and behaviour of public rooms corresponds rather well with the source domain. However, nothing can be moved from public rooms, although objects can be copied, but this is necessary to keep the interface simple. Otherwise there has to be a set of different rules depending on type of object. The main differences to both the source domain and other desktops is that clicking on a document or expertise area in the library will show the icon of the corresponding author and will not open the document or a description of the expertise area. To open a document a copy has to be created and moved outside the library. To see a description of the expertise area the member icon has to be opened by clicking. This solution was chosen to create a peopled information space, but it will result in some confusion. May be a better solution is to use another dialogue procedure to show the authors.

Private Rooms

Private rooms are just containers, mainly to be used by the members to structure their collaborative environment. Private rooms can be created by the users, they can be furnished with members, documents and CSCW tools and they can be opened and closed in the same way as public rooms. They can be also be used as interactional spaces by furnishing them with suitable tools and/or giving other members the key to the room. The member icons then indicate the members that are visiting the room, i.e. have open it. They can be used to call the members in the room, i.e. the CSCW tools are executed with the members in the room as default addresses.

They can be shared by other members by giving them a key to the room. One can also put up short messages in the room to be read by other members that have a key.

We think that this solution corresponds rather well with the source domain and that it is a useful way to both represents collaborative groups and to share this representation. It should be noted that distributing a private room to other members by giving them a key still represents a way to structure the collaborative environment not to form a team or to have a meeting.

The main problem concerns access control. We use the metaphor to give someone a key to the room. A member distributes keys by asking other members if they want to use the room. A positive answer will then result in that the same room pops up on the desktop. As in the source domain that person then has all kinds of access rights. According to our opinion the consequences of this solution is best solved by the users themselves and not by built-in rules. In addition there are other more easily solved differences to the source domain. For example the command for having a new room in CoDesk is “create”, but in the source domain rooms are not really created, but put to disposal or one gets access to a room etc.

The latter problem could easily be solved by chosing better terms. The usability of the chosen solution of sharing private rooms and of the access problems has to be tested with real users in real situation before any conclusion can be made. However the problem is the focus of on-going discussions.

Offices

Offices are mainly used to show the status of the members. The system displays some indication of on-going activities on the corresponding work station. The owner of the office can use a busy lamp to indicate that he or she want complete privacy, is busy or welcome visitors. In addition the system indicates if the owner has logged in or not. The owner can also leave short messages on the office door, as “back in a moment”. To visit an office ones opens it and can leave messages to the owner, can indicate that he or her wants to make a visit when the owner has time and can let the system inform when the owner loggs in. The office window also displays some of the CSCW tools that the office is equipped with as a telephone and a picture phone. The visitor can then see if the owner uses these tools and can use these tools to call the owner directly. The visitor can also get a description of the owner by clicking on the owner icon. Offices are opened and closed as the other rooms.

This object maps well on the source domain. Most of the important features of real offices have been represented and there is no other discrepancies that the ones caused by the general differences between the source and target domain discussed above. It is a very powerful object in CoDesk and its use could be extended in different ways. For example the user could create own hallways to structure offices into real or virtual departments.

6.4.3. Assessments of key tasks

In addition the assessments made above the target and source domain are here compared in certain key tasks as proposed by Carroll et al (1988). As there are many ways to accomplish a task in the source domain, this assessment is done by comparing the tasks in the target domain with *one* possible way of accomplishing the same tasks in the source domain (the one that has the best mapping). Just a few important tasks will be investigated as the assessment made above already has identified the main problems with the room metaphor. The tasks to be assessed are "to visit an office for discussion", "to structure a collaborative work environment","to have a meeting" and "to call one or more members.

Visit an office for discussion

Target domain:

- open the office hallway
- open the office by knocking at the door,
- observe the status indicators and messages
- call the owner by clicking on a communication tool
- start the discussion.

Source domain:

- go to an office
- observe the office to see if the owner can be disturbed
- knock on the door
- enter
- start the discussion.

Metaphorical mapping:

The mapping is rather straightforward with exception for the calling procedure. However, there is one minor difference that could be of some importance. In the target domain the office has to be opened to see the status and tools but not in the source domain.

Structuring a collaborative environment

Target domain:

- create a private room
- devote it to some task
- drag members, tools and documents appropriate for the task to the room.
- optionally, give the members the key to the room

Source domain:

- find an empty room or a desk
- devote it to some task
- place the tools and information needed for the task on the desk or in the room
- give other cooperating persons access to the room

Metaphorical mapping:

The mapping is straightforward, although this is rather unusual way of accomplishing this task in the source domain.

Call a team member

Target domain:

- select the tool
- click on the member
- start communication or click on the tool in an office and start communication
- optionally, add another member by clicking on that member

Source domain:

- start the tool
- give the address of the recipient
- start communication
- optionally order a group call

Metaphorical mapping:

The mapping is rather simple. The only difference concerns the way of addressing, but that should not be confusing. On could also add the possibility to select an address from an address list.

Having a meeting in a team

Target domain solution 1.

This solution assumes that there is a private room shared by the members of the room ie they have been given a key to it.

- inform the team members about time and place by mail or a message in the room
- equip the room with appropriate tools and document, if necessary
- open the room when the meeting starts
- check the presence of the members as indicated by their icons
- call the members
- start the meeting

Source domain solution 1:

- inform the team members about time and place e.g a conference room
- equip the meeting room with appropriate tools and documents, if necessary
- go to the meeting room

- check the presence of the members

- start the meeting.

Target domain solution 2

This solution does not assume that there is a private room shared by the members of the room

- inform the team members about time by mail or a message in a room
- call the members using suitable tools
- start the meeting

Source domain solution 2:

- inform the team members about time
- call the members using suitable tools
- start the meeting

Metaphorical mapping:

The mapping is straightforward in both solution 1 and solution 2. The most important difference between the source and target domain concerns asynchronous meetings. Such meetings does not really exists in the source domain, while they are important in the target domain.

6.5. Conclusions

This assessment of the room metaphor used in CoDesk, both points to the power of this metaphor and the problems of modelling the interface so that the metaphorical mapping is simple and straightforward. However one main problem in this kind of assessment is to model the source domain. The model used in this chapter is very tentative and should be elaborated before more assessments are made.

The methods used for assessments have indicated aspects of CoDesk that is not consistent with the source domain, aspects of the source domain that do not exist in CoDesk and features of CoDesk that are consistent with the source domain. In addition a lot of design changes have been proposed.

Most features in CoDesk could be expected from knowledge about the source domain. The structure of objects, their basic features and how they are handled corresponds rather well with the use of rooms in the real world. For example, access rights are handled in the same way as in the source domain.

Examples of features of CoDesk not existing in the source domain are expertise areas. The users then have to learn how this concept is represented and how the corresponding object should be handled and CoDesk should be designed in a way that supports this learning. Another example is the way of selecting an address by selecting a member icon. However, this should not be very confusing.

The most important example of features of the source domain that user may expect in CoDesk are conference rooms. CoDesk should then show in a very clear way that “to be shared” is a feature of a private room and that there is no separate object “shared room”. Another example of such inconsistencies is related to how objects in public rooms are handled. For example, they are not moved as other objects but copied. One way of solving this problem is to have a specific command “to copy an object from a public room” instead of a drag&drop procedure.In addition some of the terms used should be changed to support a correct metaphorical mapping.

Next, CoDesk will be redesigned according to the result of this assessments. The redesigned CoDesk will them be put to user testing to reveal other kinds of metaphorical problems in understanding and handling CoDesk.

7. Conclusions

Steve Benford

University of Nottingham

This chapter briefly summarises the first part of the Deliverable by reviewing the previous six chapters and discussing the relationships between them. It concludes by identifying future work to be done within four main areas: immediate issues for the COMIC spatial model; broader issues for the design of collaborative virtual spaces; prototyping work; and tensions between the virtual space approach and other approaches such as behind multi-media conferencing systems and media-spaces.

7.1. Summary

The first part of this Deliverable has presented results from the first year's COMIC research on cooperative work within large virtual spaces. As might be expected, the previous chapters have presented a broad range of work spanning review of previous systems, identification of requirements and development of early models and even prototypes. We begin with a brief summary of the major issues addressed so far. This is followed by a discussion of future work, both in terms of immediate next steps and also longer term objectives.

The focus of the work so far has been on spatial approaches to CSCW, whereby groups of people are situated within some kind of computer sustained space and use spatial properties to control interactions with each other, with artefacts and also to aid navigation and exploration. The use of spatial metaphors within CSCW and related disciplines is not a new idea and consequently chapter one provided a review of the most relevant previous work with particular focus on development and use of the "rooms" metaphor. Perhaps the most significant conclusion from this review is that previous work has not actually considered virtual rooms as true spaces within which occupants have positions and through which they can move. Chapter two built on this conclusion by identifying a set of high level requirements to be considered when designing more powerful virtual spaces for interaction. Typical issues included developing mechanisms for conversation management in virtual space; determining appropriate body images for people and structuring space to increase legibility and to promote cooperation. Of these issues, support for conversation management would seem to be the most central to CSCW as it concerns modelling and supporting interactions between users. As a result, this was chosen as the starting point for the COMIC modelling work.

Chapter three looked at how people interact through operations on "shared material" and examined current user interface approaches, in particular the concept WYSIWIS and the use of Video as a communication medium. It highlighted a number of problems in this area including lack of a common spatial frame of reference, lack of appropriate embodiment of users **and unflexible and rigid communication channels between the users.**

Chapter four introduced the COMIC spatial model of group interaction. In essence the model introduces a small number of key abstractions to support the flexible management of conversation and other interactions across a range of virtual spaces. These mechanisms include **aura**, a subspace which bounds the presence of an object in a given medium and which therefore acts as an enabler of interaction. Aura collisions are the starting point for all subsequent interactions. Two further concepts are then introduced to allow objects to control mutual (non-symmetrical) levels of awareness within space and hence drive interactions across different media. These are **focus**, a subspace within which an object directs its attention and **nimbus,** a subspace within which it directs its presence. The provision of both focus and nimbus allow both parties to influence their awareness of each other and hence supports the key goals of autonomy and symmetry in conversation management (in contrast to floor control techniques such a token-passing of the use of a chair-person). The model also defines **adapters** and **boundaries**, objects which amplify or attenuate aura, focus and nimbus and so make the model more flexible and extensible

This chapter also described some possible applications of the model including multi-user Virtual Reality and Populated Information Terrains. and the chapter finished by relating the model to previous work in the Social Sciences and Computer Science. More specifically, it considered the relationship between the spatial model and previous studies of cooperative work and also emerging work on distributed processing platforms.

Chapter five commented on the general use of metaphor in the design of CSCW systems, pointing to its advantages, but also offering some warnings relevant to future COMIC work. This was followed up in more detail by chapter six which analysed the application of the "rooms" metaphor in designing the user-interface to KTH's Knowledge Net system.

7.2. Future work

We divide the discussion of future work into four topics: problems with and extensions to the COMIC spatial model of interaction; other research issues to be addressed in the design of collaborative virtual spaces; prototyping and demonstration work and resolving broader tensions between the COMIC approach and multi-media and media-space systems.

7.2.1. Questions for the spatial model

We are convinced that the current spatial model provides a useful basis for further work. However, a number of immediate and perhaps difficult issues have already come to light. These need to be addressed in the next stage of COMIC.

- At present we model interactions among dyads in terms of aura, focus and nimbus. It is not clear that this is sufficient to model interaction within and between larger groups. How might the notion of group aura, nimbus and focus somehow emerge from those of individuals. What might such concepts mean? How can they be modelled? How do they affect interaction? Do we need to scale up from the present individual orientation to a more group oriented view?
- Chapter four also introduced the idea of boundary objects and their possible effects on transit, aura, focus and nimbus. Chapter four gives examples of adapter objects. What is the relationship between these two concepts? More specifically, is one subsumed by the other. To go a stage further, it is clear that any object can act as a boundary (consider a line of policemen in crowd control). Are amplification and attenuation of aura/focus/nimbus really basic properties of all objects in space (analogous to the way that all objects are affected by and affect gravity in physical space)? Also, how do we model the effects of multiple adapters on an object? Are they additive? Do they cancel?
- The spatial model currently concerns mutual awareness between objects in space. How does this relate to awareness of events? For example, does a change to an object (e.g. modification of some data) have a focus or nimbus separate to that of the object itself? Do different events result in different levels of awareness (e.g. deletion of an object causes larger ripples than reading it). This issue clearly relates to the strand four work on awareness within the shared object service.
- How should the spatial model address temporal issues? One approach might be to include time as another dimension in the system. Objects could then have a temporal nimbus or focus governing when in time they were directing their presence or attention. Another temporal issue is that of presence or attentiveness. For example, if an end user becomes inactive after a period of time, should other users become less aware of them (e.g. their body image begin to fade away)?
- We might benefit from a more powerful modelling technique for the spatial model. Chapter three presented some candidates for a diagrammatic notation. We might also develop a more formal modelling technique (e.g. some kind of field theory whereby aura, focus and nimbus are modelled as fields across space).

7.2.2. Other issues for collaborative virtual spaces

Beyond the above direct issues for the COMIC spatial model, there are a whole host of further issues to be considered in the design of large collaborative virtual spaces. Many of these were identified in chapter two. We briefly summarise the major ones below.

- What kinds of spatial structures will best promote and enable cooperative work? At the most fundamental level this raises the question of whether virtual spaces should be closely aligned to real-world metaphors or whether we should follow the more abstract route employed in the Populated Information Terrains work. It also introduces the notion of "legibility", a concept from the discipline of city planning which refers to the inherent graspability of a given space. How can we make our virtual spaces more legible and hence more usable? Can we borrow real-world techniques such as providing landmarks and accentuating districts, nodes and pathways? How do these ideas apply to more abstract spaces?
- Beyond legibility, how can we support users in navigating virtual space and in finding each other and relevant activities? Can we employ techniques such as maps, signposts, trails, tours, histories and guides?
- Virtual spaces may, in some sense, provide a direct visualisation of organisational context for cooperative work. How does the spatial model relate to the work of strand 1 in COMIC? Can we use existing organisational databases to manage virtual world information?
- How will users tailor virtual environments? What tools can we provide to manage changes to body images, spatial structures and boundaries? A related issue involves the persistence of such systems. In order to be believable virtual spaces must be persistent. This implies that construction and tailoring must take place within a working and running system. Put another way, it may not be sensible to take a virtual space off-line every time we wish to amend it in some way. The idea of changing a system while inhabiting it raises some serious technical issues.

7.2.3. Prototyping and demonstration work

Year two of COMIC should see the beginning of prototyping work where a number of demonstrators are produced to show "proof of concept". There are several possibilities to be considered. Extension of the DIVE system to implement the full spatial model and to show working boundaries and adapter objects. Extension of the CyCo and/or Knowledge Net systems to show the spatial model applied in a two dimensional context. Use of Dimension International's Superscape product at Nottingham to provide fly through simulations of a range of collaborative virtual spaces. Implementation and comparison of different approaches towards building PITS (see chapter three) using DIVE and the World Toolkit at Nottingham in conjunction with various databases and information stores including relational databases, object-oriented databases, Directory services and document stores (e.g. USENET News or Gopher/World Wide Web). These possibilities require further elaboration and evaluation.

7.3. Tensions between the COMIC approach and other work

We conclude by examining some broader issues concerning the relationship between the COMIC work and other approaches. In particular, we highlight some tensions between the "virtual space" idea and its leaning towards Virtual Reality and work on multi-media and media-spaces. These issues could easily provide the basis for extensive debate within the CSCW community over the next few years. We should stress, that we raise these issues in order to stimulate this debate.

Firstly, it is not clear how the spatial model relates to the wealth of CSCW work on multi-media conferencing systems and media-spaces. The medium of video offers a number of advantages: it is here and now; it provides realistic images of users; it can be used to enhance awareness. It also suffers from a number of problems. For a start there is no underlying model driving the image displayed. Instead video is simply and crudely a series of bits shunted down a wire. This means that manipulation of images is extremely difficult (e.g. "I want to look like this " or "move that table over there"). Building a flexible CSCW system using video might be likened to building a drawing tool using scanned images. Furthermore, video does not provide multiple users with a common spatial frame of reference, resulting in some difficulties (e.g. where is "over there"? who is next to who in a conference?). The same argument can be levelled at media-spaces. Although they serve to link up separate physical spaces (e.g. a number of offices), they don't quite create a common space. Some of the resulting problems were pointed out by Bill Gaver at CSCW'92 (e.g. limited fields of view and fixed viewpoints)[1]. The Virtual reality approach is to provide model-based animated images, a common space and multiple, mobile viewpoints. The problem here may be a lack of realism. Will people accept the kinds of crude body images used in DIVE and similar systems? Of course, it is probably a mistake to treat video transmission and animation as mutually exclusive options. The two might easily be combined, as in existing examples of texture mapped video where a video image is wrapped around some object in an animated environment.

A more significant issue might concern the relationship between embedded and immersive technologies. On the one hand media-space work aims at providing naturally embedded technology in a ubiquitous computing environment. In this case, the technology should blend in with the users natural environment and there are strong possibilities for inter-working between computer and non-computer based technologies. On the other hand, the most extreme case of Virtual Reality involves complete immersion. Here the user enters a computer generated world which aims to replace their natural environment. Beyond the discomfort of donning the currently uncomfortable and cumbersome headsets, will users find immersion acceptable? Well, it may depend upon the user. Perhaps a skilled designer might be willing to be immersed in order to work with a fully interactive 3-D design; but would an office worker for every day work? Of course, immersion is not the only possi-

1 [Gaver 92] William W. Gaver, The Affordances of Media Spaces for Collaboration, In Proc. CSCW'92, Toronto, ACM Press, 1992

bility and several VR systems also run in a desktop (i.e. screen-based) mode. However, the fundamental tension remains between enhancing the real world and entering one generated by the computer.

In spite of these unresolved issues concerning the relationship between the COMIC spatial approach and other work in the field, we believe that COMIC has made a significant contribution to the use of spatial metaphors in CSCW systems. We argue that recognising that users need to be directly embodied within a common navigable space is a critical step towards exploiting people's full range of spatial skills in future systems. Furthermore, we propose that the spatial model, particularly the concepts of aura, awareness, focus, nimbus, adapters and boundaries, constitutes a novel and powerful approach towards building these systems. In particular, we believe that the spatial model will enable us to take full advantage of emerging technologies in the area of virtual reality and harness them to effectively support people in working together.

Part II
Shared Artefacts

8. Paperwork

John Hughes and Val King
University of Lancaster

This chapter provides an outline of the social organisation of paperwork. It draws on a previous study of an entrepreneurial firm showing how documents are inextricably part of a working division of labour.

Initially, some general but socially derived properties of documents are proposed. The argument, however, is that understanding documents as social constructions involves the detailed explication of their use within a particular domain.

8.1. Introduction

One of the major challenges facing CSCW is to bring descriptions and analyses of the social organisation of work to bear on the design of CSCW systems. How are these to be mapped onto system designs intended to support collaborative activities? Many of the problems involved here are, of course, the concerns of the various strands of this project. This input is concerned to present the outlines of a sociological approach to document use in work activities. It is not written with any specific CSCW application in mind, though some general remarks are offered at the end of the paper. The immediate relevance of this input has to do with discussions on 'common artifacts', though this is not addressed directly, and the wider concern about the social organisation of work activities.

It would be hard to underestimate the importance of documents in human affairs. They feature in almost every human social activity. Much of the early socialisation and education in modern society is concerned with learning to produce, manage and use documents. Much work is explicitly concerned with the production of documents of various kinds: the accountant is concerned to produce a document which records the financial status of an individual or a company; the secretary produces letters, the social worker records, the solicitor contracts. Much work in a variety of settings consists of working with and through documents in such a way that it could be said that the work consists of producing, managing and preserving documents. There are, too, activities in which documents play a key role in such things as denoting status and/or skill, such as a driving license, a certificate of proficiency, a contract, and so on. In addition, and especially with respect to particular types of document, there has grown up an extensive set of procedures for recording, managing and indexing documents, such as the ISBN number for published materials, filing systems, catalogues, version management systems, and so on. The production, the maintenance and organisation of documents is an endemic feature of human affairs.

Not surprisingly in view of the above, in computing many of the applications that have been successfully developed have also been systems concerned with the production and the management of documents. The obvious example here is that of word processing, but also includes the spreadsheet, the drawing programme as well as indexing and retrieval systems, and data bases. The speed and the capacity of the computer have both been directed toward the production, reproduction, the storage and the management of documents as one of the most widespread and successful of computer applications: an importance that, no doubt, motivated the appellation of “The Document Company” for Rank Xerox’s publicity. Most of these applications have been developed with the single user in mind, although with the advent of networking and distributed systems, the sharing of documents at a distance has become a reality, and it is no great feat of prediction to suggest that CSCW system will offer, as an essential feature, the joint distanced production of documents.

One of the objectives of CSCW systems is to provide support for cooperating individuals who are *distant* and *time* dependent (Johansen, 1988). In addition, as Schmidt and Rodden (1992) note, it will also be important to recognise the significance of *domain-specific mechanisms of interaction* and a *repertoire of unversally applicable techniques of communication*. Thus, a CSCW platform should support both CSCW applications as well as single-user applications. Such a system should exhibit the following generic properties:

- support informal interaction in order to articulate distributed activities in dynamic and complex settings;
- support information sharing and exchange;
- support distributed decision making;
- support coordination and control protocols;
- support domain directories.

Of course, realising such a platform in ‘real world’ applications is a long way away and in need of the kind of basic research exemplified in this project. What is clearer, however, is that such systems should, as far as possible, use, if not emulate, some of the interactive properties of the social organisation of activities in order that systems are not merely ‘user-friendly’ but ‘use-able’. In particular, and again drawing on Schmidt and Rodden (1992), reflecting the seamless way in which ‘formal’ and ‘informal’ interactions are achieved, as well as the ways in which individual and collaborative work is woven into a set of activities. It is in this respect that studies in the social organisation of work assume some strategic prominence.

What follows are some remarks on the social organisation of documents. It attempts to identify *from a sociological point of view* some of the generic properties of documents as well as emphasising the importance of examining, in detail, their use within particular domains. In this respect, it tries to identify some of the 'context-free' properties of documents as well as some of their 'context-dependent' properties (Sacks, 1963). A case study will be used taken from an ethnographic study of an entrepreneurial firm (Anderson, Hughes and Sharrock, 1987).

8.2. Documents from a Sociological Point of View

Without going into anything like an extensive exposition of the available sociological perspectives, it is important to outline, very briefly, the point of view which informs what follows. The focus is the *social organisation of work activities* as these are intersubjectively experienced and understood by the parties to those activities. The social organisation of work is understood to be organised, as an ongoing and practical matter, by those parties. Accordingly, the sense, the intelligibility, the rationality, the accountability of the artifacts, the events, the activities within a work setting, are matters *achieved* by the parties to the work. The analytic task, accordingly, is to describe, as rigourously as possible, the social organisation of work in these terms. Further, this is a task which must begin not from generalised and abstracted descriptions of work and its activities, but from the detailed examination of these within their natural settings.

The above is, of course, only a very brief characterisation of the approach used (but see Garfinkel, 1967; Sharrock and Anderson, 1986; Button (ed), 1991 for fuller expositions). An important point to note is that the above is a statement of initial presuppositions for inquiry and not in any sense a theory of interaction. However, it does have one immediate and important implication. The properties of artifacts, including documents, are treated, as a matter of methodological stipulation, as *socially constructed* in and through the intersubjective understanding of members. In this respect, members' understanding of what things are is contextual and situated.

In light of the above, the first point to note about documents is that in people's socially organised lives they do not typically appear 'as documents' but as 'essays', 'contracts', 'draft articles', 'a profit and loss ledger', 'a technical drawing', a 'rail ticket' and so on. In other words, descriptions which are related to the *socially defined purpose* for which they are intended; descriptions, that is, related to their socially organised situations-of-use. In other words, documents are 'objects' whose character has to do with the socially situated purposes of use. And, as such, evoke and depend upon the contextual, commonsense and largely taken for granted practical knowledge which gives them their sense within a set of socially organised activities. Thus, such descriptors are not to be understood as 'merely' alternative ways of describing a more generic and abstract entity, a 'document', but is to take cognisance of how and in what ways they are socially constructed as 'objects' within practical activities as these are commonly understood. That is, they are descriptions of documents tied to the nature of the activities within which they appear.

Closely connected to the first point is that such descriptors are standardly available as part of the language. It is this which enables them to function as intelligible descriptions, though not necessarily correct ones, for a communication community.

In a word, they are public. Documents, to use this term with the caveats noted earlier, are *massively public*. This is their point. Documents are produced for specifiable and specified others and their 'sharedness' a function of their sociality.

A third feature is that documents can be both *normatively and pragmatically regulated in their availability and accessibility*. By the former is meant that documents can be surrounded by rights and obligations as to their access and their use. These rights and obligations can be highly variable in their formality and in their nature. The purchase of a book gives the purchaser some, but not unlimited, rights over its use. In this case, the rights pertain to content of the book not its materiality. Other documents, such as legal and medical records, are typically restricted to authorised users. Others, such as personal letters and diaries, are regulated by informal norms. Yet others are in the gift of the author who has the right to nominate other users. By pragmatically available is meant that the accessibility is largely a matter of relevance and opportunity. Thus, and typically, public notices are written and displayed 'for anyone who may be interested' or 'for anyone for whom the information is relevant'. The distinction between normative and pragmatic regulation is a rough one only and, in practise, such features are likely to be interwoven. However, the main point is that documents constitute socially constructed artifacts which are massively publicly available and socially organised and regulated in their access and availability.

A fourth feature, and one which is intimately bound up with their public availability and their information carrying property, is that documents are standardised productions. That is, they are *formatted artifacts*. The format can be understood as a set of procedures for producing and using the document (Ackroyd et al, 1992). Formats are embedded instructions for the creation and the use of documents and, once again, can be more or less formal, more or less standardised, more or less precise in their specifications. A spreadsheet, for example, is a tight systematic organisation of instructions for ordering and manipulating information. The format is the procedure for using the spreadsheet. A letter, on the other hand, can be less systematic in its format, though this does not mean that there is no structure to its organisation.

Formats are conventional and knowledge of the format essential to knowing what the document is, what it 'stands for', what it means, how to use it, and so on. Knowing how to use a spreadsheet, to return to that example, means knowing how its data is arrayed in a vertical and horizontal manner, knowing how to move within the rows and the columns, and so on. Such knowledge of formats can, of course, be highly specialised or, as in the case of books, letters, newspapers, etc., widely available.

Finally, and a property deriving from the massively public and standardised qualities, documents are traces. They can be used to reconstruct activities, agreements, judgements, states of affairs, intentions, and more, since their character is intimately part and parcel of our commonsense knowledge of persons, activities and social structures. Documents have, we might say, properties of *representation* and *documentation*.. That is, they allow for the construction of records, annals, memorabilia, reminders, and more, which can all be used as the basis for inferences about activities, persons, and events.

The above can be construed as an outline of the context-independent properties of documents in the sense that they are properties independent of specific and particular contexts of use. *They are not independent of their socially constructed character*. In other words, the claim is that these properties are generalised descriptions of social practices and conventions and, as such, do no more than point toward the need for the detailed examination of appropriate domains of use.

Turning to the context-dependent character of documents brings to the fore matters to with the instantiation of the above properties in particular work settings and occasions of use. It is with this in mind that we present two extracts from the study referred to earlier concerning the production of 'accountants' objects' (Anderson et al, 1987). By this we mean, the production of documentary 'objects' which are recognisably and visibly understood as products produced for, by and for the use of, accountants.

The original research was an ethnographic study of the top level management of a medium-sized, multi-site 'fast food' company that operates over 50 sites in leisure centres, shopping malls, airports, and so on, and employs, at the time of the study, over 500 persons. The thrust of the investigation was a topic largely unstudied by sociologists, namely, the business life of the business executive. The study itself pursued a number of themes all of which expressed the endemically social character of economic activities. The first section we have selected for discussion here concerns the production of documents, in this case 'accounts', to support the executive's organisational monitoring and decision-making. In this respect, and especially in the domain of work, the documents figure as part of what can be termed a *socio-technical production system* (Trist, 1971). This notion is meant to capture the mutifaceted character of a working environment in which both the social and the technical aspects of a production process figure as interacting subsystems.

Creative Accountancy

Sociological interest in capital accountancy begins with Weber (1978) and was, for him, a form of monetary accounting distinctive of rational economic profit-making. It is a means of assessing and validating the opportunities for profit and the success of the profit-making activity. Central to this process is calculation; calculation of the risks and chances of profit prior to the actual action, and, at its conclusion, verification of the resulting profit or loss. It is such estimations which are the *sine qua non* of business management and of the capitalist economic system. The outcomes of the system are the balances, forecasts and accounts used in business decision making and recording. It also results in documents of various kinds, such as memos, summaries, accounts sheets and so on, but the basis of the system, what makes it work, is the knowledge brought to it by those who make it work as a system of calculability. Although it could be said that what is created is materialised knowledge in the form of documents, memos, account's sheets, and so on, this is not its point. The knowledge deployed is technical knowledge of standardised accountancy practices. It is this knowledge which makes the system work, and what it works upon, as a system of calculability. It is knowledge which is taught, credentialled, and widely transferrable between organisations

This technical knowledge is socially organised. The materials to hand, and to which the knowledge is applied, the files, the data sheets, accounts, and so on, are all themselves products of socially organised accounting work. As such, they are accountable in two senses;

- the documents, the findings, materials, and so on, are widely used within the business. They are available to Directors, administrators, and managers as objects for debate, discussion, argument, defence, interrogation, and so on. Much of the business talk within the company is about these objects.
- the materials are 'displays' (Lynch, 1985) of the methods used to produce them. What anyone can find in them will depend on interpretation of the displays. Running one's eye down the print out, going through the file returns, tells one very little unless one knows what to look for. That is, knowing what the 'object' represents and its implications for *this* firm means knowing not only about this firm but also how to read the 'objects' as displays of the procedures used to produce them *for this firm*.

So, the 'technicality' of this particular socio-technical system is defined by the interplay of accounting practices, operational procedures, organisational routines, and commonsense methods of interpretation.

The point of any system of capital accounting is to arrive at some efficient and reasonable estimate of 'how things stand' in the business. It is not that any old estimate will do, but has to be one which results from the correct application of the rules to the activities in view. This means attending to the criteria for correctness and what is required for their satisfaction. What constitutes, for some specific occasion, correctness, allowable error, the margins of calculability, for example?

If we try to look at capital accounting as an empirically observable set of activities, as the operation of a system of calculability, the following features emerge.

- The knowledge which anyone working with the system possesses and uses is a locally organised corpus (Pollner, 1987). It is a body of knowledge which makes itself available to hand and within reach to those in the setting. While the procedures may be adaptations and variations of practices used in all accounting systems, they have been customised to suit local needs and organisational characteristics of this company.
- As a corpus of knowledge it is unavailable as a collection of abstracted cases and idealised procedures. No one could put together a complete list of what they know about the various sites of the company and their peculiarities, nor when this knowledge is to be applied as a set of general guide lines. Each site has to be treated, potentially at least, as a 'one off'. The primary orientation of estimations for each site is that the system of calculation should be applied as consistently as possible to all like cases. The knowledge is put to work in

achieving, first, the match of like with like and, second, the consistency between them.

- The use of this knowledge is designed to be followed. Given the wide circulation of the products of the system of accounts, any application of the system must allow others to follow it to see how the product was arrived at, why the figures 'came out' as they did, and hence how far, if at all, they need to be taken into account, relied upon, ignored or whatever.
- The products have a non-definitive character. Others can take them away for their own use, check them over, reconcile them with alternatives, make inferences on their basis, fit their procedures into other procedures, calculate them all over again, and so on. These uses cannot be constrained by the procedures which produced the objects and a variety of possibilities must be allowed for.

In the next section the focus is on just one set of procedures, namely, *documentary coordination and condensation* done by the Senior Accounts Processor as part and parcel of her routine work

The Management of Facuality

At the end of each financial period, that is, approximately every 4 to 6 weeks, the Directors meet to assess progress and determine future courses of action. Available to the Board is a set of Management Accounts for the period just ended and the year to date. The Accounts Processor's task is to build a summary paperwork picture of the financial state of the company's retail outlets. The work consists in making the Company available for the Directors by turning the local economic activities of buying and selling food into a series of accountants' objects which can then be subjected to accountants' work.

Every stock-taking fortnight, each of the unit managers is required to fill in two sheets: a stockcheck sheet and a stock-purchase sheet. Both are broken down into columns which the manager fills out. The stockcheck sheet is a computer print out listing all the items bought centrally for the units. Items not included on the list are added by the manager. The manager fills in all purchases, transfers, present stock and usage; figures which are keyed into the file update at Head Office.

The stock purchase sheet is a similar list of items arrayed by day. The manager lists supplies against the days on which they arrive. The Accounts Processor checks these figures against the delivery notes forwarded by the unit managers. Where there are discrepancies she amends the sheets in accordance with the delivery notes. At the same time she keeps a check on the prices of fresh meat and vegetables. Other prices are standardised and checked by the purchasing section when invoices arrive. This whole procedure takes one week. It is carried out while the figures on the stockcheck sheet are being keyed in.

When the files have been updated, a printout for each unit is extracted. This is the processing sheet which the Accounts Processor works on. The columns are checked row by row against the documents already processed. Where differences occur, then amendments and recalculations may be required. Invoices may be called for and, in some cases, managers required to check further.

The figures for each period are summarised on a stock-taking results sheet. These compare food and liquor costs against the notional targets of 30% and 40% of sales. Variations from these targets are the subject of decisions at the Board meetings.

All the information concerning the profit and loss of each unit is summarised on one unit profit statement showing sales, labour costs, and food costs. Unit profit allows progress against budget to be assessed.

The above is the overall structure into which the work of the Accounts Processor fits. The essence of this work is *modal transformation*. Figures are ‘picked up’ from one set of sheets and transferred to others. She takes someone else’s ‘output’, that is, the managers’ fortnightly sheets, and turns them into another’s inputs, namely, the Management Accounts and summary sheets. One set of materials is turned into another set so that others can work on them. However, this transformative work is unprescribed. The Accounts Processor has no particular targets to achieve, no results she has to come up with. Whatever the figures turn out is what she produces.

Her transformations provide a connection between the ‘over the counter’ activities of the various units and ‘managerial decision-making’. Sets of figures, reports, documents, statements, and so on, are turned into a coherent, formatted, systematic, easily described, read, summarised, visible ‘at a glance’ depiction of how things are going in the company. What this involves, first and foremost, is the physical coordination of documentation. She works down the columns looking from one sheet to another, backwards and forwards, checking off the numbers as she goes. The delivery notes are prepared in a day by day sequence so that when she goes through the purchase sheet she will have a bundle that is physically manipulable. The stock sheets are also all filed together unit by unit, as are the stock purchase sheets, the processing notes, and so on. When invoices are requires from purchasing, these are pinned to the delivery note to which they apply. The coordination of these physical objects on her desk is achievable only by keeping together the things which go together. Each stands for a particular way of characterising the objects which she has to produce. Her transformation of them involves not a synthesis, not a selection, but an amalgamation.

Although the transformation takes time to produce, it is treated as atemporal. The figures and the accounts refer to how things were at least 2 weeks ago, but are treated as representations of how things are. They exist in what Raffel (1979), in talking about clinic records, calls the ‘permanent present’. They have a fortnightly sense of now, a sense which is seen but unremarked upon, known but irrelevant for the purposes for which they are constructed. To those engaged in measuring the Company’s profitability, there seems no other sensible way of dealing with their operational and organisational contingencies other than freezing activities in fortnightly blocks. This is the level of accuracy that is all that can be practicably asked for.

This brings us to the question of the production requirements for modal transformation. What does the Accounts Processor have to work on? What must she produce? There are some important features which are visible here:

- the *formatted character* of the Management Accounts. Whatever she produces will be fitted into that format, appear alongside and be used in conjunction with other formatted depictions.
- the *perceived priority* placed upon only some of the possible measures which could be derived from the products she comes up with. Profitability is measured by closeness to the notional target proportions of sales set for food, liquor and labour costs. Not just any set of figures can be used and not just any procedures for amalgamating them can be invoked. The logic she applies, and the steps she takes to work her way through are fixed by the conceptualisation of the Company's activities built into the system of calculability. It is for this reason that the figures and sheets she produces are 'accountants' objects'.
- the work is *product guaranteed*. The managers fill the sheets in as a requirement of their jobs; the delivery notes are assembled with the invoices as a requirement of invoice payment. As a result, she will always have enough resources to 'get the figures out' and, hence, provide some account of how things are going. The physical presence of the sheets and invoices in the office provides her with enough to bring the projected course of action to a conclusion. Though for this reason or that some of the figures may be shakey, for the purposes for which they are used they will always do.
- the transformation is the *assembly of an order*. She does not need to look beyond the documents she gets to produce the depictions she does. If there is a query it is from the paperwork that it is resolved.
- the tasks are organised a *principle of opportuness* (Sharrock and Anderson, 1987). The whole monitoring system is possible because of the centralisation of accounting and purchasing. Much of the coordination of activities she is able to achieve is possible because she is 'going through the paperwork' at the same time as assembling the accounts.

The work of the Accounts Processor is a sheet by sheet, column by column, row by row, modal transformation of one set of accountants' objects into another. This is managed by the physical coordination of bundles of documents, documents in files, and documents being worked on. As she works through the lists she leaves a trail of markings so that anyone who knows her routine can come to the files and see where she has got to. The records and documents she produces are a permanent account of the sequential organisation of the tasks comprising account processing in this company. The beginning to end trajectory of the whole sequence is achieved stage by stage, one step at a time, by making sure that everything is to hand that is needed, getting the documents if you need them, checking the files when necessary, and so on. It is in her orientation to these things as seen but unremarked, necessary features of her work tasks that the ordinary orderliness of the Accounts Processor's working life consists.

What has to be done to make a system of calculability work is the production of sets of calculables, objects to be manipulated in calculations. The company's paperwork system is designed to produce calculables as part and parcel of the other functions performed, such as paying invoices and wages, controlling costs, monitoring cash flow, and so on. The production of calculables is, therefore, just one of the outcomes of the socio-technical system of capital accounting. But it is an outcome which can be incorporated into the essential feature of all capitalistic endeavours, namely, the determination of profitability and the decision making based upon that criterion. The modal transformation achieve by the Accounts Processor's work of documentary coordination and summary facilitates the process of interpretive calculation without which such decision making would not have the character it has.

The Accounts Processor's work, though performed largely by her, is part of a sequence of activities within which it is situated. The documents flow into the Accounts Processor, are transformed, and passed on. The work which produces them is part of a division of labour.

8.3. Documents within a Division of Labour

If we turn to the organisation of work around the Purchase Ledger Desk in the company, we can see how the division of labour operates within an environment which is saturated with information. As can be seen from the diagram (Figure 8.1), the office consists of a number of positions occupied at any one time by a particular person. The distribution of positions is work specific and is the sedimentation of the local production practices for invoice processing and cheque paying. From the point of view of the accomplishment of such tasks, the work appears as an impersonalised stream of tasks in hand, tasks completed and tasks to be done. Within the bounds of competence, it is of no matter who carries out the work. Rosemary, Eileen, or Amanda could just as easily as Renee fill out the journals and make out the cheques. In this sense, the personnel are locally interchangeable and, often, 'cover for each other' when the need arises. Nonetheless, the task differentiation and specification for the work is seen in terms of 'decisions-that-I-can-make' and 'actions-that-I-can-take' and those which others deal with. The division of labour is organised according to a principle of *egological determination* (Natanson, 1986). That is, from the point of view of the actor in a division of labour, working through the endless stream, getting things done, means 'doing-what-I-do' and passing on tasks to others so that they can do what it is they do.

The organisation provides an institutionally located and, thus, socially available allocation of activities and tasks. The major line seems to be oriented around the centrality of the individual and bounded by the horizons of their task performance; that is, its egological character. These spheres of operation vary from those whose boundaries are permanently open, under review and hence always near to hand, to those which considerably removed, closed and taken for granted.

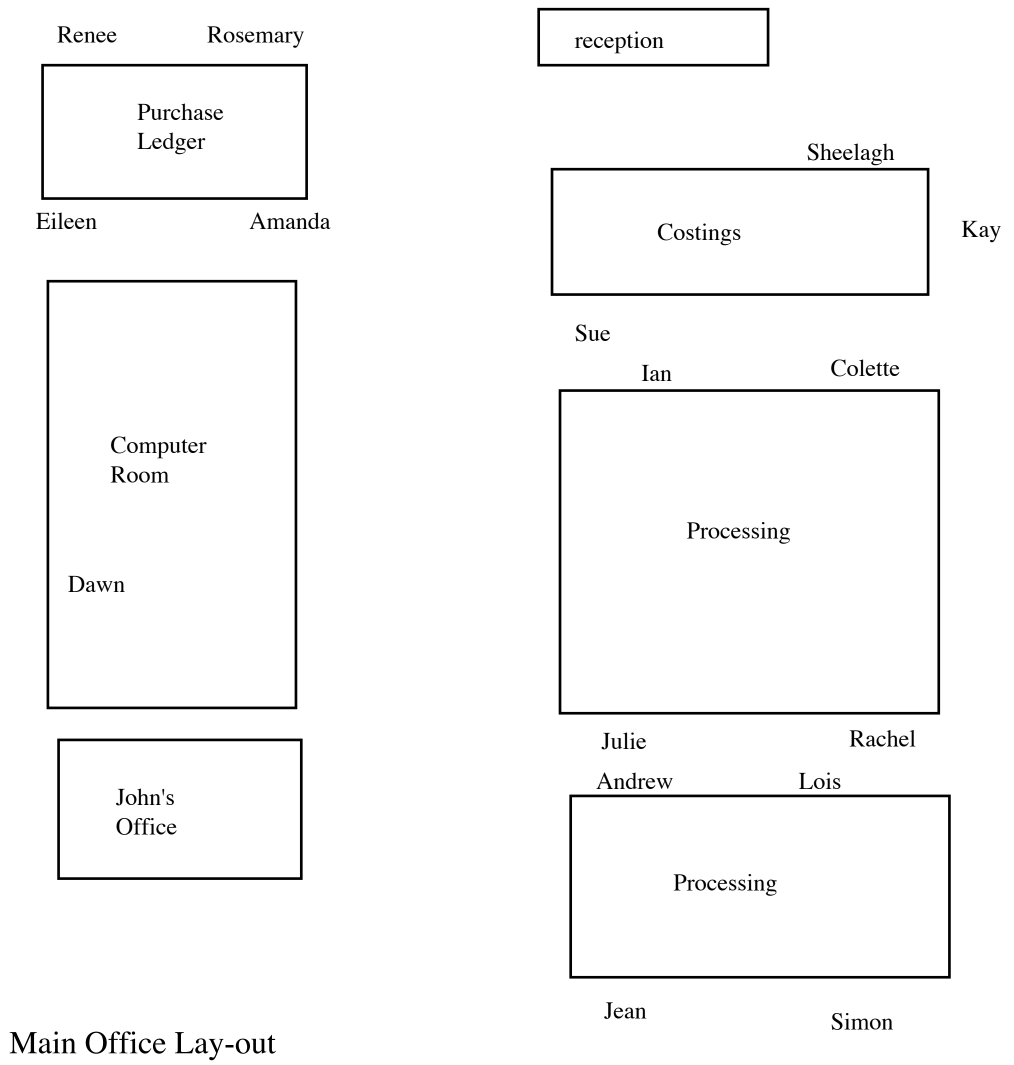


Figure 8.1: Main Office Layout

For the accomplishment of any tasks, some aspects will be problematic. Other will be treated as matters beyond inquiry. For Sue on the Costings Desk to be able to check the prices on the invoices does not require her to worry about the postal service, the grounds on which food and non-food invoices are separated, nor what happens to them when she has passed them on. Once she has finished with them and sent them to Purchase Ledger, they are no longer her concern. Such matters can, of course, be brought under scrutiny. From what 'anybody' knows about the company and how similar enterprises operate, it is possible to discover the reasons why things are done in the way that they are. So, when asked to 'pull an invoice' which she has already passed on, from what she knows about the system and how it works, Sue can find her way through the division of labour at the Purchase

Ledger desk and locate the item. Similarly, if required to check an invoice with a supplier, she has little difficulty in working her way through the accounting procedures of another company. She achieves these by treating the division of labour as a distribution of locations for the accomplishment of activities. What she has to find out is where, in the processes for the production of financial accounts, the work she wants to query gets done. What this *reciprocity of locations* is built upon is a presumption that some solution to the problem will have been provided, someone will do the work in more or less known ways, just who and where can be discovered if required.

Parallel to this presumption of the reciprocity of locations is a matching *horizontal distribution of knowledge and interests* which provides a working and workable basis for the termination of interest and inquiry. This is rooted in this commonsense precept that, practically, not everything can be questioned at once and not everything needs to be questioned at any one time. Dawn, for example, keys the codes and values into the computer without needing to have any detailed knowledge of the routines which produce the management accounts, the computer printed cheques and bank transfers. It is enough for her to know that somehow the work which she carries out enables these processes to happen. But it is crucial to the routine accomplishment of her weekly and fornightly round of work that she orients to the tightly specified timetable of the accounting fornight. By knowing what the timetable is and whereabouts they are within its prescribed order, she can juggle the batches of invoices, wage sheets, stock control sheets and other inputs in order to ensure that all the fornightly routines can run on time.

The egological principle underpinning the working division of labour generates and provides a solution to the coordination of tasks; a matter of 'gearing into' the division of labour' (Gurwitsch, 1979). This is because the division of labour specifies which tasks one has to embed within one's activities and which are, so to speak, institutionally taken care of. This involves anticipating how the institutionalised structures will work. Memo's can be put in pigeon holes in the certainty that they will be eventually read and acted upon. Invoices can be left in baskets for others to pick up. If, later, they have gone from the basket, then they have been collected. Such routine operations of the division of labour provide organisationally specific ways in which those within it can call up, gear into it, and make it work for them. Dawn, for example, has to check if the coversheet for each bundle of invoices has been completely and correctly filled in. She has to set the number of invoices in the bundle right at the start so that the routine loops the appropriate number of times. If the box is empty or the count is wrong, this snarls up her work. On the other hand, should the routine's internal monitor reject an invoice, she merely returns the invoice to the processors and holds the bundle until it is returned. It is a feature of the institutionalised character of accounting procedures that there are enquiry procedures for just this kind of contingency. Dawn knows this. But sorting out the problem is someone else's work. Gearing in, then, is a means of ensuring the smooth performance of the flow of activities by ensuring the fit of one's task performance into that of others and by carrying out 'running repairs'.

Activities are not only organised egologically, but are also zoned according to organisationally relevant dimensions of space and time. Fig 1 showing the floor plan of the office shows the accounting relevant distribution of niches. The work is clustered around types of audit check. Other distinctions as marked by the Divisional structure of the company are not visible here but are of interest only so far as they bear upon the performance of the tasks in this office. It is an environment of paper processing. To those who know the office and its work, the layour recapitulates the division of labour in that the sequencing of activities can be reconstructed from the mapping of who is where by tracking an invoice through the system and the floor of the office. Thus, when locating where a particular item might be, or some action takes place, a glance around the office suffices as a reconstitution of the organisational plan.

The accounting work is also tied to the technology available. This consists of a filing system which utilises box files and print outs as well as manually operated desk calculators. What any person can do at any one time is constrained by what they can get on their desk, who has the files out and where, what can be photocopied, what can be written alongside what, whereabouts the information is stored and how much trouble it would be to get it. On the other hand, fax machines and telephones reduce the importance of proximity. The work carried out is both constrained and freed by the technology in use. This is particularly the case with Dawn. The routines she follows are fixed by the accounting fornight and the information processing routines of the computer. When she is inputting data, she clocks through the windows which the routines provide whether or not they are required for a particular invoice. Her worked is fixed by the features of these routines. She can only move backwards and forwards within windows in certain ways and can only call them up in a particular order. However, she can call routines up and juggle with the types of in-puts she makes in whatever ways she needs to keep the work flowing.

From the point of view of anyone immersed in a *working division of labour*, activities move through fore and background according to the principles of egological determination and structures of relevance. The organisation of activities is not a fixed, given, system-specified phenomenon but an outcome of the routine coordinating work which those working within it perform. Because such coordination is routinised, taken care of as a day's work, working in the division of labour renders the coordination of activities unproblematic and invisible. It is because the production process 'works' that the problem of coordination of activities disappears. However, the division of labour can be a members' methodological trouble and can be constructed 'in flight'. In both these instances, the point of determining a division of labour is a matter of working within a context of justification. That is, the efficiency, the rationality, the effectiveness of a given organisation is a matter of determining the appropriate criteria for measurement by reference to the division of labour itself. Such criteria are inextricably tied to the context in which the division of labour is being constructed.

The Inter-actional construction of a division of labour

When looked at from within a production process, a division of labour, and particularly when looked at in terms of how the features of the objects it produces are recognised and deployed in the taken-for-granted way that they must be, an object passing through it is seen and treated as a *stratified record* of the work producing it (Garfinkel, 1967; Raffel, 1979; Lynch, 1985). The object displays the locally organised construction of the division of labour for all who know how to see it. In the office, the production process is concerned to produce ‘accountants’ objects’ and is, essentially, a division of paperwork labour.

The invoice as a stratified record of work

The company receives invoices in a constant flow but pays them only fortnightly when a series of computerised cheques are run off. As far as food processing is concerned, there are two forms of invoice, Food and Non-food which are treated in different ways. The reasons for this are managerial, financial and organisational. Given the nature of the business, the major proportion of invoices are for food and drink items. One person can deal with these efficiently if they specialise. Given also that food purchases are centralised through main suppliers, the supervisory role can be carried by this person, who can also check that the appropriate prices are charged. A cross-check of food purchases is also obtained through the unit’s fornightly returns on usage. Cross-checks do not occur for non-food items and knowledge of the appropriate prices are distributed among the management. Circulation of the invoices is required for checking purposes alone. Such circulation does, however, allow for the supervision of spending at the respective units; the means by which the Directors keep their fingers on ‘what is going on’.

Any invoice has a circulation life within the company, that is, the length of time it takes to process and pay it. The shortest this is likely to be is a week; the longest 6 or 8 weeks. During this life time it passes through a number of hands and across a number of desks. The invoice contains a record of its own passage displayed on a date stamp. Invoices circulate in bundles which start life as ‘what is in the post’ and accumulate into a ‘day’s worth’ after coding by Purchase Ledger. From there on they accrete into larger sized bundles depending on how quickly they are processed into later stages. The most likely place for this accretion to occur is in Lawrence’s office, the General Manager, where they can sit for up to a week.

Each stage of the process is dependent on completion of prior stages for its own completion. It is scheduled to ‘fit around’ the fixed points of the computer input, the weekly wage payments and the fortnightly stocksheet runs. This means that invoice processing is a continuous, fitted-in-where-it-can-be matter for Dawn in the computer room. There is a constant backlog on processed invoices to key in.

The rationale for having a separate invoice processing function is, first, it provides a degree of financial control and, second, allows efficiency of effort by freeing management from the task of checking bills, etc.. As a succession of tasks to be done in a series, the processing of an invoice has itself to be fitted into the daily and weekly routines of the those that deal with them. No one deals with just invoices.

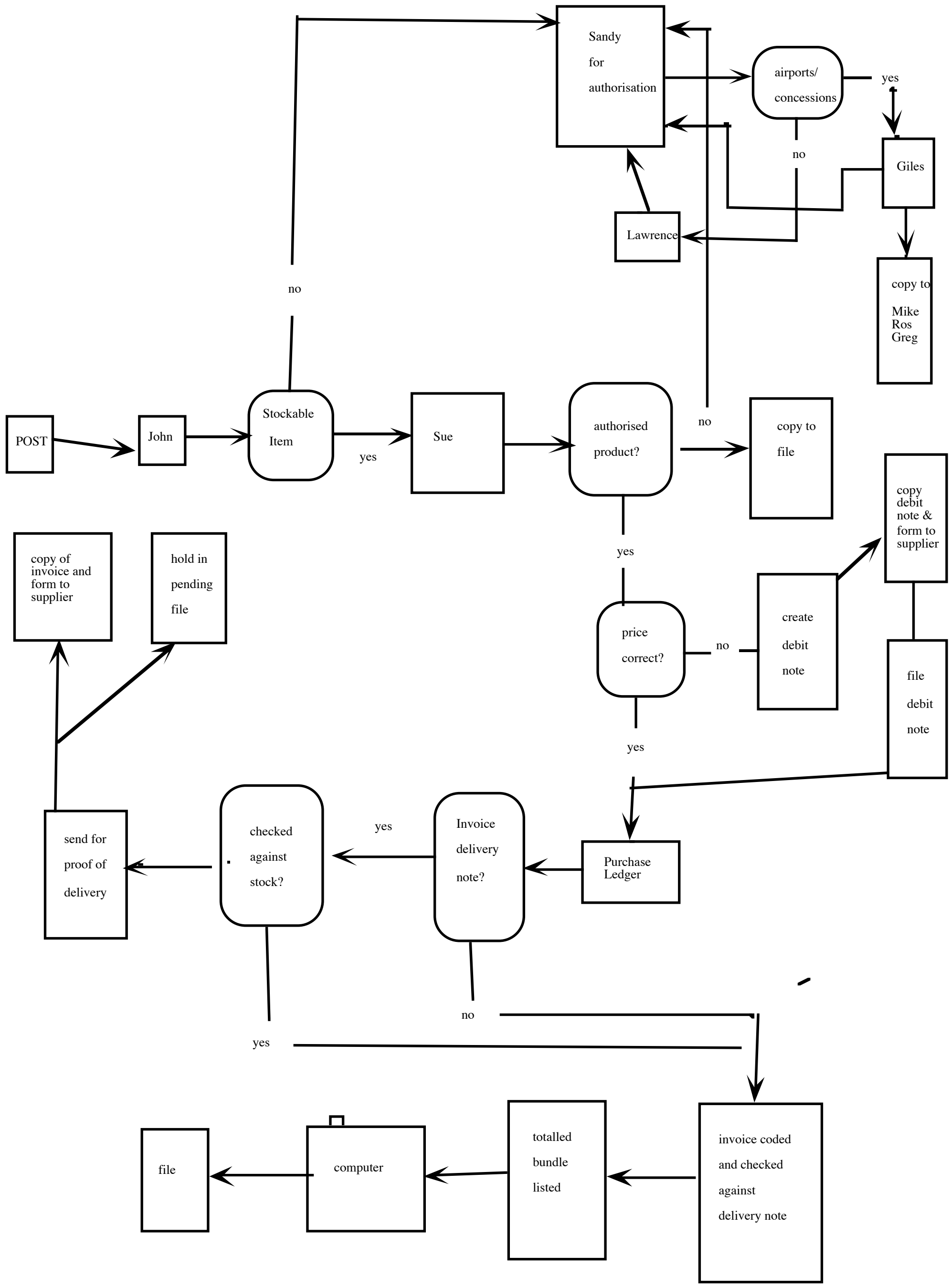


Figure 8.2: Flow diagram for non-food invoices

The first stage is when the invoice arrives at Head Office it is stamped and passed onto Deborah. She separates the food and non-food invoices and pulls out any she might want Sandy to see. The food invoices are sent to Sue for coding. Non-food invoices go to Rosemary. The coding procedure is via a stamp on the invoice providing a number of boxes to fill in with codes. These codes file the information on the invoice in the computer. SUPP NO, for example, is supplier number as specified in the computer's list of authorised suppliers. Thus ASB 001 means the main account with Associated Biscuits. A designation is given for the division of the company against which the invoice will be charged. Rosemary also checks the totals. Once it has been coded, the non-food invoice goes to the Divisional director for authorising. Authorised invoices are returned to Sandy who passes them onto Lawrence. He signs, or queries, them and passes them back. He generally goes through them once or twice a week, largely when he has nothing better to occupy himself with. The invoices are returned to Purchase Ledger where they are bundled with a cover sheet into 70s and sent to the computer room. The cover sheet gives the date and the codes of the invoices in the bundle. After being typed into the computer they are filed.

The only difference from this is where the invoice is urgent. In this case a manual check is made out. This requires that Rene make a cash book entry, a journal amendment entry for the computer and to process the cheque and get Sandy to sign. Generally, all electricity and gas bills are paid this way and it involves a lot of extra work.

As it moves around the paperwork socio-technical system, the invoice acquires a record of the work done upon it. From the moment it arrives and is date stamped, all production work leaves its mark upon it, either in spaces provided within the date stamp or as appended comments, memos, queries, questions staped to it or stuck on it. In this sense, the orderliness of this record on the invoice is a representation of the orderliness of the work tasks performed. The representation appears as a tick against items that have been checked, question marks against those which are unknown, the initials in the various boxes, and so on. To anyone coming in to an invoice at any moment of its path through the system, the recording on the invoice of what has been and what has not been done, builds up to a *statified representation* of the sequence of stages it has passed through and the actions taken in regard to it. Since this sequence and these activities are standardised, a glance at the stratified record is enough to be able to tell what has happened, where things are up to, and what the possible problems might be. Knowledge of the accounting practises in general and any particular realisation of them are mutually explicative.

For those engaged in the production of 'accountants objects', the orderliness of the record is the orderliness of the tasks. One goes proxy for the other. As the boxes are filled in, as the correct codes are written in, the amounts checked and the authorisations given, the normal, unproblematic routine working of the system reproduces itself. Work upon the invoice is, therefore, a distinct sphere of operations for those whose tasks it is to ensure that the paperwork is completed properly. Its horizons, its internal organisation, and its structures are given to them as local and contextual knowledge about how things are done in the company and what, from the invoice, one can say about what has been done and what yet needs to be done. Learning to read an invoice as a record of its production work is learning the paperwork division of labour.

8.4. Some Implications

A first response to note, and emphasise, is that the study summarised above is not simply a study of documents but a study of work activities and how they are organised. Documents are intimately tied to social activities and this is what a sociology of documents will amount to. That documents are tied to, in this example, the activity of producing accounts is what gives them their distinctive character and sense as socially understandable 'objects'. Knowing what the accounts mean, what they 'stand for' is knowing how and why and for what reason they are produced. Documents, we might say, are 'activity-bound' for their intelligibility, but not so in all the uses to which they may be put.

The second point is that although the paperwork described here is a highly routinised activity using tightly formatted procedures and displays, much local and specialised knowledge is required to do the work 'correctly'. Their production is not 'merely' a mechanical process but also an interpretative one involving, for example, knowledge of the sites, invoicing procedures, 'what seems correct', and so on, in order to get the overall accounts to the level of accuracy required 'for this organisation at this time for the purposes for which they are required'.

A third point is that the work is scheduled into a series of stages which is a working division of labour. Thus, although rarely do the personnel work together on their tasks, the individual work is embedded in an organisation of work which depends upon the performance of others. The work is, to use an earlier phrase, organised on egological principles. Nevertheless, the work retains its massively public character in a number of ways. Not only are the documentary materials used available to others, and intendedly so, the work done, not done, yet to do, is available as it accumulates the stratified traces of the work. The flow of the documents, particularly the invoices, around the office provides not only a record of work done, but also informally provides a sense of what is going on in the firm, especially for the Managers. It is clear that much local knowledge is employed not only in the doing of the work, but also in understanding what the work produces. Much of this knowledge, as in the case of the senior managers, a function of the flow of documents itself. The fact that the documents circulate as part of a division of labour provides not only the means by which the various transformations of the materials are achieved, but also an indication of 'how the firm is performing'.

What follows by way of implication for possible CSCW applications is difficult to determine in the abstract. Also, it needs to be noted that many of the documents that can be used in work activities, accountancy being perhaps a salient example, are themselves the product of computerised processes. Indeed, much of the information used by the Accounts Processor in the study referred to is stored in the firm's computer. Much also depends upon the work activities, hence the type of document(s) that are involved, for which computer support is envisaged as well as the nature of the collaborative activities involved. It is clear, however, that the kind of properties that paper provides within a setting such as the one described, even though many more of the processes could be computerised, may be difficult to preserve.

9. If All The Worlds' A Stage, Are Documents The Script?

Val King and John Hughes

University of Lancaster

This chapter is a response to the request for scenarios of work activities and their settings which involve the production and the use of records. The settings chosen are a social work agency, the police use of records, clinic intake records and, finally, records within laboratory work.

The chapter emphasises those aspects of document use which are relevant to the design and development of effective systems technologies which support work activities where records are a major aspect of that work.

9.1 Introduction.

This paper is a contribution to some of the work in Strand 4 on shared objects. It draws out some implications of the socially organised character of documents focusing particularly in documents which are used as records within an organisation. It extends the work presented in COMIC Working Paper, 'Paperwork'.[1]

This particular working paper is relevant to that aspect of the research concerned with a Shared Object Service (SOS) (see Lancs-4-2) for CSCW applications. Such a shared object service would be needed to support CSCW applications and provide appropriate mechanisms for representing people, groups and projects and, essentially, effective means of sharing materials.

The rationale for examining the social organisation of record use is to inform the design of the SOS in order that it may be sensitive to the 'real world' characteristics of record use and yet offer the advantages of an electronic system, particularly, flexibility, shareability, distributiveness, as well as the well known advantages of electronic storage and retrieval.

The analysis draws upon published ethnographic studies of work settings in which documents of various kinds figure as a shared resource among parties to the work activities concerned. The work settings were:

- social work

1 A companion COMIC document to this is 'Paperwork' by Hughes and King (1992).

- police work

The intention was to select different work settings but no claim is made that these are representative of document use within all organisations, though we do suggest that they indicate some of the generic properties of document use within work activities. Moreover, the analysis presented here is general and there remains a need for a much more detailed study of document use within particular work settings.[1]

The studies used and referred to as resources for this working paper are as follows:

Ackroyd, S., Harper, R., Hughes, J.A., Shapiro, D., and Soothill, K. (1992), *New Technology and Practical Police Work,* London, Open University Press.

Wattam, C. (1992), *Making a Case in Child Protection,* London, Longmans.

9.1.1. The Analytic Focus

It is a commonplace to note that many work activities are directly concerned with the production, completion, storage, retrieval and use of records of all kinds.[2] One need only be reminded of such prosaic uses as bus tickets, forms of all kinds, letters, notices, certificates, licenses, and many more, all produced as part and parcel of various kinds of activities. Such records, it needs to be stressed, are not to be considered as adjuncts to the work concerned but integral to it. That is, and to stick with a prosaic example, the ticket that is issued to a passenger by the bus driver when collecting fares, is not an incidental of the work of 'collecting fares' but integral in the sense that 'the ticket' stands as a record of a passenger having paid the correct fare, that the ticket documents this fact, and its issuance part of the work of transporting paying passengers. In a similar vein, though perhaps less prosaically, the memos, the letters, the forms, the records which are standard features of much of organisational life are integral to the organisation's work.

This means that records are also part of an arrangement of interdependent activities. For example, the bus ticket 'documents' the relationship between fares and passengers and, presumably for managers, is a means of checking the probity of employees, a measure of workload, the rate of revenue generation, and so on. In other words, records are rarely isolated documents but implicated within manifold and interconnected arrangements to do with work within an organisation setting.

From the point of view of those 'doing the work', records are both a product and a resource for this 'doing' and this is an oriented to feature of the everyday

1 It should be noted here that Risø is currently carrying out a study connected with COMIC on the computerised management and production of documents for manufactured artifacts. In addition, the project on software development, funded by the HCI Initiative and involving the Departments of Computing and Sociology at Lancaster, and the Department of Sociology at Manchester, will also provide useful material in this connection.

2 It is this fact which has stimulated some of the earliest and most familiar applications of computer technology in, for example, data bases, spread sheets, and so on, innovations which themselves owe much to the first modern mechanisation of data analysis from records, the Hollerith machine.

working life. In other words, record use is to be understood as a *using* within the sociality of work and its organisation.

The analytic focus is on work as a collection of socially organised activities, that is, as a set of activities produced and reproduced by parties to the work itself as these are intersubjectively experienced and understood by the parties to those activities. The social organisation of work is understood to be organised, as an ongoing and practical matter, by those parties. Accordingly, the sense, the intelligibility, the accountability of the work, its artefacts, its events, its personnel are matters achieved by parties to the work.[1]

The above is a statement of the initial presuppositions for inquiry and is not in any sense a theory of interaction. Its main implication is that the properties of artefacts, including records, are to be treated, as a matter of methodological stipulation, as socially constructed in and through the intersubjective understandings of members.[2]

In 'Paperwork' (COMIC-LANCS-4-4 and previous chapter) it was noted that within ordinary social life documents do not typically appear as 'documents' but as 'essays', 'contracts', 'draft articles', an 'invoice', a 'technical drawing', and more: labels which are tied to the socially defined activity in which they have a place. Descriptions, that is, which are tied to their socially organised situations-of-use and which evoke the contextual, commonsense and practical knowledge which gives documents their sense within a set of socially organised activities. Such descriptions are not to be understood 'merely' as alternative ways of describing a more generic and abstract entity, a 'document', but is to take cognisance of how and in what ways they are socially constituted as objects *within* practical activities. In other words, the sociological study of records and documents is the study of work and its activities.

Earlier it was noted that documents, records, are integral to work activities. However, it is not the case that the work itself and its activities can be 'read off' from the document or from the record without further ado. Documents, though tied to the notion of 'providing evidence for', in the way, for example, that a train ticket is 'evidence for' having paid to travel on the relevant train, only in general terms can the work which produces that document, which supports it as a 'trustable' item, as 'evidence for having paid the fare', be inferred. Documents, records, and the like, are often glosses of the work which goes into their production. It is in knowing what the record represents which provides for its situated use within the setting concerned. 'Knowing what the record represents' means knowing about the work that produced it, what it means within *this* activity, within *this* organisation and how it might be used. It is to know, for further example, what it means that X

1 This is a brief characterisation of the Ethnomethodological approach to the sociological analysis of work activities. See Garfinkel (1967), Sharrock and Anderson (1986) and Button (ed) (1991) for fuller expositions.

2 It should be made clear that this is not an argument to the effect that documents, artifacts, events, persons, etc. have an entirely subjective meaning such that everyone sees things in their own distinctive way. While there is a sense to this, what is emphasised here is the *intersubjectivity* of social phenomena which focusses on the extent to which the social world is a shared one.

produced the document as opposed to Y, where X is a head of department and Y a junior manager. It is to know what follows by way of action from a particular police report, say. It is to know, in the social work agency, how the cumulative file 'adds up' to reasonably grounded suspicion that a 'child is in danger'.

The technicalities of the particular working practices are visible in and through the local, socially organised use of documents, and knowledge of these practices. This has the methodological implication that understanding the use of records means understanding the kind of work activities in which they are embedded.

The Work Settings

As indicated earlier, the work settings in which records form a significant aspect of the work activities are diverse although this is not to say that there are not properties of records and documents which are describably generic.

What typifies the settings referred to here are:

- They consist of offices in which there is an interdependence of activities. That is, a distribution of roles and responsibilities, to borrow this language, such that tasks, jobs, responsibilities are distributed among personnel although the separate contributions have, in some sense, to contribute to the organisation's work.
- Thus, while any individual may have their own particular task and particular responsibilities, these do not encompass all the necessary tasks and responsibilities to carrying out what can be glossed as the 'work of the organisation'. This does not necessarily mean that jobs consist of specialised functions, though it may do as in the case of the police, but there is, nevertheless, a strong sense in which the work is not individual but takes on a corporate identity.

The social work setting is the regional office of a national charitable agency concerned with the protection of children from abuses of various kinds, including sexual abuse. The office consists of a small secretariat and rooms for the use of social workers who, for much of their working day, are out on calls. It also includes interview rooms, including a two-way mirror room for video-ing interviews with children. Any particular social worker may be dealing with anything of up to 20-30 cases at any one time, but at varying levels of the cases' careers. Thus, dealing with cases, though sequential, is not sequential in 'clock time'. That is, a social workers will deal with aspects of one case - maybe a visit to check that 'things are ok' - and then go on to another case to arrange for financial support, or whatever. So, in the course of a typical day a social worker will perhaps visit a number of cases of different kinds and at different stages. When they are not out on calls, they have the task of keeping their records up to date, dealing with issues arising out of their cases, attending case conferences and other meetings, writing reports, and so on.

The police work is diverse, and includes uniform patrolling, supervision, detectives and higher level management. Constabularies are divided into.geographical divisions, and, laterally, between uniform and detectives. Most 'line' police officers spend their time dealing with the public. It is calls from the public which largely determines their patterns of work. In addition, there are the typical bureaucratic positions within the force which do not involve much contact with the public. Much of police work is heavily bureaucratic involving the production of considerable documentation, from the constable's log book, to the charge sheet, to the statement, to the documentation requires for prosecution in the courts, to the typical administrative documentation to do with the management of large numbers of personnel, such as rosters, personnel information, etc.

Like social workers the day-to-day work of a police officer 'on the street' can be broken up into various activities depending on what arises during that day. Although much patrol work, for example, is routine, emergencies or incidents must be attended to as a first priority and these initiate paperwork. As indicated, police work generates documents of various kinds as a matter of course. This is especially true of detective work which is, despite popular mythology, mainly bureaucratic.

The records, too, exhibit a diversity, both of character and of function. A common sense of records is to regard them as documenting the past events or actions of an organisation's members. However, while there is some sense to this, it hardly captures the variety of uses to which records are put. Thus, although the records of a social work agency, and those of the police, do document past actions they are often 'live' records in that they are in current use and can be expected to accumulate other items as part of that record. As we shall indicate, one of the properties of records for an organisation is that they can project, or initiate, consequent actions on the basis of the current record. A summons, for example, initiates a sequence of actions on the part of the police, the courts and the person summoned. The documentation for the design of the software system could be said to serve a dual purpose: as a blueprint for the building of the system and as a record of any changes or amendments that may have been made during the course of this development process.

Although the perspective of the social organisation of work emphasises the particularity of work settings as an initial stance, from the point of view of the SOS, the aim is to look for general features of work settings which can inform the specification of a CSCW SOS.

In the next section we identify some of the general social organisational features of records. Note that within specific organisational settings, details and practices are likely to vary.

9.2. Records Speak Within an Organisation

If documents are the script for the world's stage, then their voice is found within organisations. What records mean, what they refer to, what they might indicate, what they constitute, what they are, has everything to do with their place within some organisational setting and its activities. Records are, we might say, organisational objects and, as such, represent and display organisational activities. This is what they record, namely, activities which are relevant to the organisation's business. However, this is not intended to mean that records only make sense within a particular organisation. While it is the case that many records are for the use of a particular organisation, they do have trans-organisational relevances. The classic case of this is, perhaps, police records which can be transferred into 'objects' used by the legal system. Similarly, medical, social work, financial records, among others, can be used by other organisations though, typically, not in their 'original' form but usually in some transformed fashion.

In some respects the above point seems obvious in that the very notion of record, of a document, is to make publically available in some way, what it is the document records. It stands, that is, by its nature as interrogatable by others - indeed, it could be argued, that this is the purpose of a record. However, the point we want to emphasise is that the movement of a document or a record between organisational settings, so to speak, is not merely the movement of pieces of paper, or even electronic text, but is a process of the transformation of organisational objects which brings other relevances come into play.

That is, a record can serve many purposes. The humble bus ticket, as pointed out earlier, can be interrogated as a means of checking the honesty of the driver, an idnciator of work loads, the relationship between revenue and work load, etc.. However, and to repeat the point, the issue is the relevances that drop into place to constitute the different usings.

There are, of course, records which are specifically designed for inter-organisational commerce, invoices, cheques, purchase orders, letters, being among the most common.

A final preliminary point though difficult to express with much brevity. In much of what follows we use the term 'organisation' as a generic term to refer to organisations such as firms, businesses, schools, universities, research labs, government agencies, social work agencies, and so on. That is, to identifiable, socially recognised ensembles of persons who can be conventionally described as working for 'organisation/firm/company/ X'. So, when we speak of someone acting in an organisational capacity, we intend nothing more than that they are 'representing', 'acting in', their identity as a member of the relevant organisation. We make this point to avoid any implication of reifying organisations as if they acted like persons. The point of view taken here is that only persons can act; organisations act only metaphorically. We will consider some features of the social character of records.

9.2.1 Some Social Features of Records

The following list of the social attributes of organisational records attempts to identify some of their more salient features. It is important to note that these are not necessarily discrete and independent features but are interwoven within the patterns of socially organised activities within an organisation. Thus, what they identify are often overlapping, mutually supporting features of records rather than single, distinct and exclusive properties.

Records are part of a socially organised pattern of work

The production, maintenance and use of records is itself a work activity and one often intimately tied to the work that the records display. Thus, in social work providing records relevant to a case, such as recording the results of a home visit, an interview, a medical investigation, etc., are the direct responsibility of the particular social worker as part of his or her job. Similarly, medical records are produced as an integral part of the doctor's work as they are of police work. In the case of software design, the records are produced as integral parts of writing the code, doing the design, and so on.

In some cases the production of records is itself a separable activity as the direct responsibility of some person(s) in their organisational capacity. Indeed, some organisations have record departments whose responsibility it is to keep particular classes of records.

Records as representations of organisational objects and action

As we have said, records are part and parcel of the activities of an organisation and represent those activities. Many documents are explicitly understood to 'stand for' the activities of an organisation - a rail ticket, a letter from a Managing Director, an official notice from a government body, an advertisement, and so on. Profit and loss accounts are taken as a measure of the success, or otherwise, of the performance of a business enterprise. Records can be examined and interrogated for what they say, what they indicate, about the organisation's activities in dealing with its 'objects'. In this respect, records can serve as *traces,* as *archives,* of what the organisation did in the past and as *meters* for what the organisation is currently doing.

However, as we said earlier, records are rarely, if ever, straightforwardly records of the 'actual activities' of an organisation's personnel. Rather, their objective is to represent these in organisationally relevant ways and are, accordingly, glosses for those activities. It is this which gives the phrase 'off the record' its poignancy. They are often couched in conventional formats, using organisation styles (as in legal documents), are often summaries of 'what transpired' (as in the case of minutes), and so on.

It is probably worth distinguishing, albeit roughly, between 'official documents' and 'personal records'. By the latter we mean things like personal notes, 'post it', scribbles, and such like, that are often produced by personnel during a working day. By the former we mean the organisationally official communications and records. Sometimes the distinction is not a hard and fast one, and it certainly does not mean that from the point of view of the social organisation of work it is only the latter which are of interest. Nevertheless, the main focus here is upon the organisationally official.

It can be said that one of the purposes of records is to make the activities of organisational personnel accountable. However there are two related senses of 'accountable' here. One is its usage in Ethnomethodology and denotes the ways in which the character of the world, to use this rather clumsy phrase, is made visible for members and is, thus, a generic feature of inter-subjective understanding. The other is a more standard usage in which we might 'hold someone to account', that is, asking them to justify their actions. This latter sense is dealt with later.

Records as the sediments of an organisation's activities

This feature exemplifies the previous one. Typically records can be examined for what they say and indicate about the activities of an organisation and its personnel. In this respect, *traceability* is provided by indications of such things as *authorship* of the record. That is, not just 'anyone' is entitled to produce or add to the record but only someone, in their organisational capacity, who is *entitled* to do so. To this end, there are conventional forms used to denote who, in what capacity, and when, has used the record in some way.

This feature is supported by the common practice of storing records, an activity which, especially within large organisations, is a specialised task of such as secretaries, administrators, etc. Thus, the sedimented records become deliberate historical records which may be used in accounting for an organisations actions. Such accountability of an organisation may be prosaic in, for example, an invoice standing for what the organisation owes another, a record of what a suspect has done in the past, what design decisions were made and how they were instantiated, and so on, or, more dramatically, where fraud or misbehaviour is suspected.

Records as shared objects

By this we mean that as 'organisational objects' records are produced by and for the use of organisationally accredited persons. Records can be said to 'belong' to an organisation not merely in the sense that they are, as it were, 'owned' by the organisation concerned but also in the sense that they are *for* the organisation's personnel. They are produced for specifiable and specified others within the organisation and in their organisational capacity.

This property could be said to be *the* rationale for records, that is, their purpose is to be shared by at least some others within the organisation for the purpose of their organisationally relevant activities.

Records as normatively produced and accessible documents

Although records are produced by and for persons in their organisational capacity, access to them is normally regulated not only to 'outsiders' but also within the organisation itself. There is, that is, an organisation to the use and access of records themselves. This can, of course, vary considerable from fairly open to closed access. The latter is perhaps most pronounced in government departments which very often grade records in terms of who is entitled to read and use them. Typically within an organisation there are users who are *entitled* to use the records in organisationally relevant ways. This may not mean that 'others' may not read them. The issue here is one of 'use within organisationally relevant ways'. Thus, and for simple example, though a secretary may be responsible for typing many of the documents which enter into, say, a social work record, s/he is not entitled to use that record in the ways that a social worker is entitled to do so. This feature is, of course, intimately bound up with the kind of understanding of the organisation and the practices of its personnel which are intrinsic to providing records with a voice.

Records and procedural implicativeness

Within the organisation, records very often have a 'procedural implicativeness' in that what they record is taken as the basis for further actions, further activities. Thus, and for example, a case report within the social work agency may well form the basis for initiating further action. That is, what the record says is the basis for other organisational actions.

Thus, police records typically have a iterative quality to them which can be described as a step-by-step process in which a records transformation to the next stage is dependent on actions taken in response to the record at the previous stage.

This feature is often denoted in the name of the record which, in its organisation context, specifies an action, an activity, in the way that, for example, an invoice not only records 'goods delivered' but also specifies 'payment' of those goods as a consequent action. Indeed, it can be argued that one of the main rationales of records is their intimate connection with organisational actions, recording and specifying what has been and what needs to be done as a consequence. In this way, they serve to underpin the interdependence of activities within a working division of labour. In this respect, their relationship to an organisation's specified roles is also important.

Records as Justifications

Within the context of some work activities, records are not only implicative of subsequent actions, but are constituted as records for the purpose of the justification of actions. This is typically, though not uniquely, the case with social work and medical records which are situated within a legal or quasi-legal environment. Such records are produced 'with an eye' to the fact that they may be inspected for whether or not the correct procedure has been followed. In other words, records can have an *evidential* character to them as indicators of some features of personnel performance and, through this, organisational performance and responsibility.

In an important article, Garfinkel (1967) notes how what is generally referred to as the evidential character of records, can be an oriented to feature in the production of records which constrains their 'adequacy' as records. A consistent feature of the case records that he examined was the occasional and elliptical character of its remarks and information. He likened this to the utterances in a conversation with an unknown audience which, becauses it already knows what may be talked about, is capable of reading hints. They make sense by reading them not as actuarial records but as a therapeutic contract between the patient and the medical staff of the clinic. That is, they are to be understood as the products of a 'medico-legal enterprise' which can be the subject of legal inquiry and, therefore, designed for this eventually.

Records as part of collection

One of the features that emerges from the studies is that records are rarely isolated items but form part of a collection of records. Many records are organised into 'cases', as in social work, in medical work and in the case of some police records, which assemble the separate items of the record or file 'as belonging together'. They are about a particular organisational 'object', be it a patient, a crime or criminal, or about the production of software. Files may also be subcategorised or aggregated further.

The importance of this feature, as we shall see below, is that typically a record is an item within a larger collection which can be described, for example, as a file on X. This means that the sense of an item is, in part, shaped by the items with which it is collected.

Of course, it is possible to accumulate records in a variety of ways according to a number of categories and principles. One way in which police records are transformed is their aggregation into the Official Statistics on Crime. There are, no doubt, similar practices in many other organisations which uses records to provide summary data on organisational activities and performances.

The prospective-retrospective sense of the records

Records typically *accumulate*. Rarely is it the case that a specific size can be nominated such that it can be said that the record is complete when this size has been reached. Records typically end when the action or the activities to which they refer cease. When a patient moves to a different district, for example, the medical record normally moves with that patient and ceases to be a 'live' record for the practitioner. The record becomes an *archive*. When a crime is solved, the current police record 'goes off the books' and becomes an archive.[1] Though in this case, such records can be reactivated.

Social work files, for example, typically accumulate materials germane to a particular case as do medical records. Documents are added to the file as necessary This means that what the record shows is a matter of examining and interpreting what earlier items mean in terms of subsequent ones, and seeing subsequent ones 'in light' of earlier ones. Records acquire whatever meaning they come to possess by readers 'combinatorially' (Garfinkel, 1967) using the items in the record together. Thus, the medical record of a patient can be interrogated for its history and what this 'history' represents can be a matter of what is happening to that patient now.

---

1 In the police, clearing up unsolved crimes is a major activity, partly because it helps the record of the respective department. Sometimes suspects are offered the possibility of 'taking into account', that is, confessing to other crimes they may have committed in the hope of a more lenient sentence.

Records can change their complexion depending on the use to which they are put.

Records and the need for organisational knowledge

One of the overriding features of record use is the need for organisational knowledge in order to understand what it is the record 'stands for', what it represents. This often involves dealing with what can be described as 'omissions' in the record which can be 'filled in' by an 'organisationally competent reader' of the record. This is one of the major points arising from Wattam's study of social work records.[1] Understanding what a file contains, what it implies, what it is for, is inescapably a function of a user's grasp of the organisation and its ways. Often such knowledge is also professional knowledge; that is, knowledge which entitles and enables the relevant person be a 'professionally competent user' of the record.

However, it should be stressed that records do not 'speak for themselves'. As we indicated earlier, records have to be made sense of, and although in many cases this sense is conventional and has a 'typicality' to it, this is always a contingent achievement. This is a feature which, of course, affects all records, though the matter of 'what the record shows' can itself be the subject of specific courses of action and not only in courts of law.

Records as matters of inquiry

This point emphasises one or two of the other features noted. Records are matters of inquiry both for organisational personnel and others for whom the records, and what they contain, may be relevant. In other words, records are not unequivocal renderings of organisational activities but can be 'framed' and used in 'new' ways. They can be interrogated, organised, reviewed, etc. for what they indicate about those activities. For example, police records can be interrogated to provide management information about the performance of the organisation and its personnel. Records can be resources for determining fault or blame or responsibility.

Records as part of transformative processes

Typically records are integral to transformative processes. That is, they are used to assemble and so transform objects, persons, events into 'organisational objects' of various kinds. Thus, and for example, an 'incident report' may turn out to be a 'crime' which is consequential for police actions. The results of these actions may well produce other records about a 'perpetrator', about a 'victim', and so on. At each stage of this *transformational unfolding* what the record refers to, what it signifies, changes through its course, though retaining its connectedness to the previous records. What this 'connectedness' consists in, of course, is again variable. In the case of the Official Statistics on Crime, the connectedness is

---

1 This is also one of the major themes within ethnomethodological inquiries into the documentation of activities. See Garfinkel (1967).

essentially statistical, but it is the record's relationship to organisational actions which is the key here.

The above does not exhaust the relevant features of the socially organised character of records, nor is this necessarily the most perspicuous way of presenting them. It is important to reiterate, too, that the above features are not to be treated as discrete, unifunctional aspects of records. Rather, they are features which all records display, though depending on the organisation some have more emphasis than do others.

9.3. Manual Record Using

In this section we want to focus more directly on the ways in which records are used within the relevant organisations paying particular attention to features which are relevant to any proposals for the electronic transformation of records. With the exception of the police - and we will have more to say on this later - most of the record systems reviewed here are manual. Even in the case of the police, electronic record keeping systems are often duplicated by manual ones and, what is more, closely replicate some of the features of such systems.

By *manual system* we mean that the records are typically

- paper based
- produced by a process of writing or typing
- stored in filing cabinets and/or their equivalent
- within an office or other facility designed for this purpose

Of course, the manual character of record systems is a contingent fact of technological history. It is also a contingent and highly important feature of the prosaic ways in which records come to be incorporated into an organisation's work. Nor is it irrelevant that the overwhelmingly typical material for records is paper. This gives them properties of robustness, shareability, mobility, visualisability related to, for example, the current everyday use of records.

It also means that organsational records are often considerable, bulky and for their efficient use require organising by means of appropriate indexes, formats, places where they can be used, and so on. This gives rise to what may be characterised as an *ecology of manual record use*.

9.3.1 The Ecology of Manual Record Use

By this we mean the relationship of the records to the organisational environment in which they are situated. Much of this concerns the spatial distribution of records in terms of where they are stored, where they are managed and from where they are accessible.

Typically, in most organisations the site of records is the 'office' making use of filing cabinets to store the relevant records and files. The organising principle tends to be standardised indexes for filing, and administration by secretaries whose responsibility it is to manage the up-to-date composition of the files and their access. Larger organisations tend to consist of a number of such offices, some tied to particular personnel, others for more general use. There can also be offices whose particular purpose is to process specific kinds of records, such as invoices, statistics, for example.

Thus, for any organisational member using the files requires knowledge of the location and distribution of the relevant files; a matter of *access*.

9.3.2 The Accessibility of Records

What we refer to as the ecology of records is not simply about the location of records within an organisation but also about their accessibility. The point of records is to support the work activities of an organisation's personnel and, thus, need to be accessible to those personnel.

Once again, in relatively small organisations, the social work agency would be an example, location in the office means that they are relatively physically accessible to those who require to use them. Larger organisations present other problems in this regard since relevant records might be physically distributed elsewhere within the organisation. This problem tends to be tackled by the production of multiple copies of records and multiple storage facilities. However, in some cases, and the traditional manual systems used by the police is a case in point, relevant records may be distributed across a number of sites and accessible only by the relevant and authorised officer physically going to retrieve them.

This particular case highlights some of the normative considerations with regard to accessibility, namely, that the retrieval of a record can only be done by an authorised officer. In other words, the records themselves are *policed* as to their entitled access.

9.3.3 The Shareability of Records

As we have pointed out, records are organisational objects and, thus, produced for the use of organisational members. This means, to reiterate the point, that records are not produced as a personal *aide memoir* but as documents which may be consulted, referred to, interrogated by, discussed by, etc., by other relevant organisational personnel.

Within manual systems this tends to be done in the following ways:

- individual personnel using the record as occasion warrants according to some sequential principle, typically, first come first served, by right or by request;
- occasioned meetings using the records as a resource for the discussion;
- producing duplicate copies of the record.

In other words, there is normally no simultaneous sharing of a record except when, such as at meetings, records are consulted. In the latter case, records have to be physically relocated to the place of the meeting unless they are copied.

Thus, in manual systems records tend to be shared over time rather than simultaneously. Where they are simultaneously shared this is done, normally, by producing copies of the relevant record.

9.3.4 Standardisation and the Formatting of Records

One of the prosaic qualities of records, and related to their shareability, is their standardisation by means of formatting schemes. Such schemes can, to make use of a rough distinction, be 'loose' or 'tightly' formatted. By the latter is meant schemes for inputting records which require the inputter to work through a strict sequence of items on a record sheet. Examples of these include summons notices issued by the police although instances of such formats abound. 'Loosely' formatted schemes tend not to impose such a strict sequence of actions on the part of the inputter.

It should be pointed out that all records are formatted though not necessarily as a record. Thus, a letter from a patient or from a parent in a child protection case is formatted appropriately as a letter rather than *as a record*. It can, however, enter a record by being placed in a file. Many records are a combination of both types, such as social work and medical records.

9.3.5 Authenticity of the Record

In a number of settings, the original paper record has a particular status *as the original record*. In other words a consequential distinction is made between 'original' and 'copy' versions; consequential in that the former, for example, has a legal status that the latter does not possess. Thus, although copies may be made of the original, it is the original which authenticates the copies *as* copies.and the original as the original. There is not anything magical about this, or about paper. What is drawn upon are the manifold institutional processes, prosaic and dramatic, which underpin this trust. This is obviously important in those legal and quasi-legal circumstances where the records can be made to stand as evidence for actions taken, but, in any event, is a taken-for-granted feature of so much of record use in ways that are not so for electronic documents.

As noted earlier, there are conventional means for determing the authenticity of the record and its status. Typically possession of a current rail ticket is sufficient to certify payment of the fare, whereas using a Railcard requires a photograph and appropriate certification to authenticate this document. In a similar vein, copies of personal cheques are not normally allowable as 'authentic cheques'. Such authenticating devices, though very often conventional, are nonetheless subtle and highly varied and, of course, have bearing upon the 'trustability' of the record.

9.3.6 Archives vs Records in Use

This draws on a distinction noted earlier between an archival record and a record in use where the latter refers to records which are 'live' in that they relate to current activities of the organisation. What we are calling archival records are records that cease to be 'live' in this sense though may be kept archivally, often in case they may be reactivated. What records are kept, and for what reasons, will vary between organisations. Police records tend to be kept for very long periods, whereas other types of records may be discarded after a shorter period. It may also be the case that only certain types of records are kept while others are discareded.

Records in use may also refer to records actually being used 'at this moment' and, in manual systems means that their location is other than the place where they are normally kept, being typically on someone's desk or elsewhere. That is, in manual systems 'using records at this time' normally means they have to be physically relocated and that they are not available for someone else to use until the current user has returned them. This raises problems of the traceability of the file.

9.3.7 Holding the file 'to hand'

An important property of records in, for example, the police, social work, medical records among many others, is that of 'having the file to hand'; that is, its manual manipulability. Part of this is being able to hold all of the records pertaining to a case within the confines of a folder and being able to lay out, browse, burrow, read again, use, the whole of the file at 'one sitting' as it were. In manual systems, 'having the file to hand' can literally mean just this with all that this implies in terms of ease of 'unmediated' manipulability.

It is difficult to say more in detail about the importance of this property since it is such a familiar and taken-for-granted feature of manual record use. The fact that documents can be moved around, rearranged, inspected, glanced at, etc. with all the ease of manual use is difficult to replicate with current electronic systems. This is not to say that, with familiarity, electronic systems will also develop ease of use, but currently they do not yet possess this quality.

9.4. Discussion

Throughout this exposition it has been taken for granted that what constitutes a record is unproblematic both for the analyst and for the users of records. In some respects, for users this is not a problem since, within organisations, record producing and record keeping are specified and relatively clearly defined activities. A police officer, for example, learns as part of learning the job, what must be written, what must be produced and in what form, how it has to be further processed, and so on. Learning to produce the relevant records and to process them is indistinguishable from learning the relevant duties of becoming a police officer. Producing the relevant records is, if not quite second nature to the typical officer, is certainly very close to that. Records become *procedurally tied* to the duties and activities of being a police officer. Police work is, one might say, paper work in that few actions of an organisationally relevant kind are undocumented. What we have called the procedural tying of records to duties is also typical of medical personnel, social workers, and many other occupations. Carrying out the duties produces a trail of records which, due in large part to training, have a fairly standardised character and style.

It is also the case that police officers, to use this example once again, typically deal with *standardly formatted* documents in which what is to be recorded, in what order and in what ways is closely shaped by the category organisation of the document itself or by standardised ways of producing reports of incidents, interviews, etc. It is these properties, for example, which have been exploited in Incident Logs and CRS systems by which incidents are recorded by leading the investigating officer through the record itself as it is completed. Thus, case files normally consist of collections of standardly formatted documents.

In social work, however, the question of what constitutes a record, and a record of what, displays a slightly different character. What the record shows is more typically a matter of explicit inquiry in which it has to be decided whether or not the record constitutes an *adequate rendering* of the phenomenon in question. Does the record provide *sufficient evidence* that child abuse is taking place as opposed to child neglect.[1] Determining this is typically a matter of 'case conferences' in which a number of relevant parties, often including medical practitioners and police officers, discuss particular cases using the record as a resource for these deliberations. Such deliberations then become a 'matter of record' in being added to the record of the case.

A process of selection is also visible in the use of social work and clinic records though, in these cases, the issue is less one of determining which items of a record should be literally discarded, but more a matter of determining what an item means in the context of the whole collection of items that constitute the record file. As the file accumulates, what an item means may well change in light of new additions, and what new additions mean can be interpreted in light of earlier ones. In other words, what the record shows is a constant matter of inquiry, negotiation, discussion, deliberation, and so on, on the part of organisationally accredited persons. Moreover, such files are less standardly formatted compared to records produced by police officers

The point to emphasise here is not so much that records and what they say is a matter of interpretation - this would apply to all records and documents - so much as a question of how this procedure enters into the very construction and use of the record itself within a particular organisational context and what actions might ensue as a result of what the record says. In social work, in police work and in other cases, the record can be examined as an account of organisational actions which are *consequentially connected* and which is exhibited in the record itself. In other words, the records can be inspected for the way in which they *account for* the organisational actions which the records report, and account for in organisationally relevant ways. Thus, and for example, within police work a record would need to provide a warrant for why a person was interviewed about a particular crime. But, as noted earlier, such accounts are very often 'glosses' of the actual rationales for the action taken'; 'glosses' which provide for *organisationally relevant reasons* for the action taken.

---

1 These categories are not merely matters of 'how to describe what the records show' but can be legally consequential in terms of what powers a social worker may employ to deal with the case. Such powers have to be justified.

One important feature of records which provides for the property of *consequential connectedness* is the 'second order' organisation of records into 'cases' or 'projects'. Records are about some organisationally relevant 'thing'. In the case of police records these may be about a person, a crime or an incident but how they become so is a matter of how the record accumulates and *unfolds* in the course of the work. Thus, an 'incident report' may turn out to be a 'crime', and a crime of a particular kind. It may also produce a perpetrator, a 'criminal', and a victim, and so on. At each stage of this *transformational unfolding* what the record refers to, what it signifies, changes through its course. However, this transformational unfolding is in reference to, and thus represents, actions with regard to 'organisational objects', be they 'child protection cases', 'crimes known to the police', 'research projects', or whatever. Such categories not only provide an organisation to the record but also provide for a 'competent user' to understand what the record is about, where in the transformational unfolding it is, what likely actions are to follow, what it means in terms of responsibilities and duties, and so forth.

What this also points to are processes to do with the *assembly of a record*. It is not the case that a record 'just happens' even though, in some cases, it might seem that material is 'just added' to the record as it occurs. A record is constructed and this process is an achievement of the parties to the work using their knowledge and understanding of the organisational procedures by which records are produced as accountable objects within the organisation itself. Not 'just anything' can be part of a record. A record has to refer to, or represent, some organisational activity and is assembled with this in mind.

Important in this respect is the way in which records are assembled through time and the ways in which this is made relevant to the organisation of the record itself, the consequential connectedness of the activities, etc. it records and also understanding the transformational unfolding of the record. Typically, records are logged by date and time as they are incorporated into the record file. This becomes a *sequential index* and a resource for interpreting the record as a representation of actions taken with respect to the 'case'. Thus, a 'competent user' of the file can use the temporal index as a way of reading what the state of the case is 'thus far' and, through this, infer as to the sequence of actions done and what further needs to be done in respect of the organisational object.

Finally, and relevant to the initial question of what constitutes a record, is the issue of a record's *life cycle*. Here one might hazard a rather crude distinction between an 'active record' and an 'archival record'. The former is a record still in the process of transformational unfolding - a case still unsolved, a child still under the protection of the agency, an ongoing project - whereas the latter is a record which has, in terms of the organisation, been 'dealt with' - a trial completed, a patient treated, a project which has been concluded, and so on. In addition, it draws attention to the temporal organisation of records in the sense that records begin, are processed and then come to an end as the activities and processes they record cease. Typically, the cessation of a file is marked in some way, for example, by 'signing off', relocation of the file elsewhere, and so on.

This brings out once again, the intimate connection between records and the organisational activities of which they are a part. As these begin, proceed and end, so does the character and the point of the record. As the activities cease, this does not mean that records are discarded. Typically, organisations retain their records, and do so for a variety of reasons. In the case of the police this is for legal reasons. In the case of medical records and social work records it is because of what Garfinkel refers to as the 'contract' between patients or clients and their professional guardians. However, such records become archives which can be inspected for a variety of purposes. In the case of the police, archival records are further transformed into official statistics which summarise the organisation's activities.

Finally, a note on shareability. Though virtually eponymous for CSCW, shareability is a complex notion especially when considered within real world contexts rather than as an abstraction. Much of what has been reviewed earlier about records instances, we would want to say, the sharedness of records and does so in all the manifold ways in which social life itself can be described as a shared experience, or as a shared world. In other words, sharedness in this sense is not some discrete, identifiable activity set apart from other social activities but is a condition of all social activities.

However, connecting this down to 'real world' activities is less than straightforward which is why, so far in this document, we have not featured this as a distinctive issue.[1] Instead we have preferred to talk about organised and interdependent work activities which are, inevitably, social and thereby exhibiting 'sharedness' in the sense just indicated. Thus, organisations are about organising and managing the interdependencies and, as inevitable parts of this, organising and managing tasks, activities, information, relationships and, of course, records and their use.

The upshot is the fairly obvious point, but one that needs to be taken with some seriousness, that 'sharedness' is multifaceted and an aspect of much of organisational life in all its everyday details. Thus, although one might think of a distinction between 'individuals' and 'groups' this does not for one moment capture anything like the necessary subtleties, details and intracacies of what these mean within particular organisational settings. These are matters to do with organisational practices. This not only makes the computational implications less than clear, but also specifies the need to look closely at the specifics of work activities within their organisational settings.

---

1 This is, of course, one of the important aspects of the COMIC project.

9.5. Implications for CSCW Systems

In this section we want to hazard some implications for CSCW system interpreting this fairly widely as equivalent to systems which support the socially organised character of work activities. This gives us the opportunity to move slowly up to the issues and also review, albeit briefly, some of the ways in which computers are currently being used to support work activities particularly in relationship to the police and identify some of the issues involved in taking further steps to support the work using computerised records.

9.5.1. Some Vague but Necessary Assumptions

One of the major problems is, of course, anticipating not only what could be done with existing technology and a minimum of reorganisation of work activities, but what might be done with new technology and a radical reorganisation of work that this makes possible. These are, of course, extreme options. Table 1 sets the mix of these general possibilities.

It should benoted that none of the cells have been completed with either existing CSCW systems or envisioned ones. There are many possibilities here but these are beyond the courage of the authors.

	CSCW System	
Organisation of Work	existing technology	radical technology
no change		
change		

Table 9.1: Options for Work Reorganisation and CSCW Technology

Of course, it is important to remember that the categories used here are both gross and very static. For example, technology new to an organisation or to a set of work activities, no matter how old it may be in technological terms, might well have a profound impact on work activities. Alternatively, it is difficult to imagine a radical technology not having a profound impact on work activities. However, though crude it does have the merit of reminding us that CSCW is not simply about technology but about the design of work itself in which computer system technology figures. It also reminds us that although it is possible to envision radically new technologies, along with new ways in which persons might be organised in their work, new technologies tend to be costly, unavailable to most people in most work settings, and often suffer from a long development time before it is ready for more widespread use. The more immediate point, however, are the assumptions we need to make about the level of technology for the purposes of this paper.

One of the first applications of computer systems was to the production, storage and retrieval of records to improve those aspects of manual systems which were relatively cumbersome:

- handling large amounts of information
- bulky storage
- speedy retrieval
- physical dispersal of records

Thus, and for example, payroll and personnel records, among the first type of records to be computerised, lent themselves readily to this technology in that they tended to be standardly formatted in terms of the information recorded, large data bases, repeatable at regular intervals, and already highly centralised. Such records tended also to be the same format for everyone on the payroll. Payments, etc. also tended to be highly procedural in that payments were the outcome of the appropriate application of standard formulae.

Such systems also tended to have a relatively large support staff inputting and amending information as necessary. Also the sharedness of such records was minimal being largely a matter of the individual and the organisation, in the case of payroll and personnel records, and had relatively little to do with the support of work activities within the wider organisation.

For the purposes of this paper, we make the following minimal and very general assumptions about the systems. They will have -

- networking capability
- relatively easy access of relevant workers to inter-connected PC's or workstations
- interactive capability

We recognise that these are somewhat vaguely stated and, even minimally, dependent upon infrastructures they not all organisations either have access to, or can afford. Social work, for example, is particularly deficient in resources for the purchase of even minimal computing systems while, at least until fairly recently, police forces in the UK have had access to reasonable resources for building and development computer systems of various kinds. The networking capability, for another difficult issue, is assumed to include communication with other organisations or access to databases which may not currently exist. There are many other issues here which, though beyond the remit of this working paper, are nonetheless vital for the development of 'real world' CSCW systems.

One of the other thorny issues is, of course, envisioning CSCW systems; that is, systems which *support* work activities. One of the distinctive features of CSCW, though not perhaps unique to it, is its commitment to examine the character of work activities first and then think about appropriate system support. In the cases used for the purposes of this working paper, it has to be admitted that the computational implications are not clear unless some assumptions are made. Here we will make a minimal assumption that the character of the work will not radically change. Whether this is realistic is another matter and, as we shall see, in some cases what could be described as a minimal technological change had significant repercussions on the work activities and morale of a group of workers.

The networking of computerised information services, either locally within an organisation or, less salient for other than the research community, outside the organisation, has transformed many aspects of the record production, maintenance and retrieval. In particular, they have offered the following potentialities[1]:

- the replacement of a manual ecology with an electronic one
- allow the simultaneous use of records
- provide access to a large variety of records

9.5.2. An Electronic Ecology of Record Use

The 'centralisation' of records within a networked computer system can have the effect of collapsing the spatial distribution that is necessary with manual records. All that is required is access to an appropriate terminal to bring the record 'to use'. The case of police records is particularly pertinent here.

Police work is particularly document sedimenting. For example, detectives spend between 70-80% of their time at the station and only about 10% of arrests are made by them. Their main responsibilities are:

- to collect information about crimes and to produce and maintain the records of criminal activities for both internal and Home Office consumption;
- to collect papers and prepare information from these records, to interview suspects so as to provide evidence for any court action which may result.

In essence, they have the task of transforming the various features of reported crime into bureaucratic phenomena. Their work has a production line quality to it involving almost every step of the way a considerable amount of documentation and record keeping. The Crime Reporting Bureau (CRS) introduced into some constabularies by placing terminals almost on every detective's desk enables them to access records much more easily and efficiently removing the need for them to travel from place to place when the system was a manual one. This not only reduces time but has increased the willingness of detectives to use the records. It has also vastly improved the access time and the variety of ways in which records can be accessed including date, location, type of crime, name of perpetrator, aggrieved persons name, items involved, and so on. Records can be accessed day or night and is not controlled by the custodians of that information.

Of course, this type of system differs from the earlier payroll models in being distributed and in summary form offers the following advantages to detectives:

- makes available records from all over the country
- does this quickly
- can be searched in a variety of ways

---

1 We call these 'potentialities' for the fairly obvious reason that whether or not they are implementted depends upon many factors, including resources, organisational policy, and so on.

- provides better opportunities for briefing detectives for interviewing suspects
- can be used as a database system for crime pattern analysis.

However, it needs to be pointed out that once accessed records become part of a manual use system in that their availability to, for example, the Crown Prosecution Service, other constabularies, etc., is typically in their manual rather than an electronic form.

Social work agencies have achieved nothing like these facilities, partly for reasons of funding and partly because records tend to have a more local relevance than police records can have. The Police National Computer (PNC), for example, is a centralised information service for all constabularies and is, in the main, devoted to providing information about vehicles and drivers. It is one of the most successful of computer installations in the police service. It is regularly used by line officers in the course of their duties and is, to repeat, a nationwide service.

However, it needs to be noted that the information 'on line' in the PNC is very standardised information relating to vehicles and drivers; information provided by the registry of vehicles through the national licensing system which also includes drivers.

It can readily be seen how this kind of technology could be more widespread and enhance the ability to retrieve records more effectively by removing the problems of multi-site storage. The problem this gives rise to is, of course, increasing the amount of information available with the consequent problems of providing maps for the information. The small office with its couple of filing cabinets is relatively simply to organise, but electronic storage, though offering some advantages where large numbers of records are involved, does have overheads in indexing, up-dating files, etc.

In the police case, this was done by creating a special group of civilian empoyees to classify crimes as they are entered into CRS. This took away from detectives their prerogative as 'keepers of the crime figures'. Though this speeded up the entry of incidents into the system, there was a period when detectives virtually 'worked to rule' because, as they saw it, one of their major skills had been taken away from them and, insult to injury, given to 'civilians'.

9.5.3. Simultaneous Use

As noted earlier, there there are a number of necessary distinctions to be made as to what constitutes the 'shared use' of records.

In manual systems, 'sharing records' is typically a matter of the physical copresence of the parties using, ' to hand', the record concerned in a meeting of some kind. Using the 'same' record without copresence, for example, two social workers wanting to consult the 'same' case but without the need to discuss it, usually means taking turns with the physical record or, alternatively, producing copies. Electronic records can, of course, be distributed, as in the police case, without the need to make physical copies.

It is also necessary to be clear as to the point of the sharing. In social work, for example, the point of meetings is to come to some agreement as to what should be done about a particular case. Such meetings tend to include various professionals and representatives of involved agencies, such as the police, probation services, and so on. The meetings are to determine what actions should follow, and by whom, to coordinate the responses and the responsibilities of the various agencies, and so on. The records serves as a resource for this. In other words, the point of the meeting is not simple to provide information but to come to some collective judgement about the distribution of responsibilities and the consequences of this determination in terms of organisational actions.

9.5.4 Access to Variety of Records

Another potential bonus of distributed systems is that they allow for the possibility of accessing a much greater variety of records than is typically the case with manual systems. Ease of access to a network capability can, as in the case of PNC, allow for using more and a greater variety of records.

There are obvious advantages in this capability for organisations such as the police which, though in the UK are locally organised, often have the need to access records from without the locality.[1] Again, PNC is a case in point, though it is not the only example from the police service. However, it is not clear that the same advantages would accrue to social work. Access to legal databases, up-to-date records of social service, financial entitlements, and so on, might be a useful aid but would carry a large overhead in inputting and updating such information. The data on PNC, it should be noted, is produced as a byproduct of vehicle and driver licensing.

The above potentialities say very little about the design of a CSCW SOS which would, at a minimum and in light of the assumptions made earlier, have to support more or less existing work activities.

A word or two of qualification is necessary at this point. 'Supporting work activities' does not mean that these activities should remain entirely unchanged. One of the purposes of introducing computerisation is to enhance work activities either by providing more facilities for the work activities or by enhancing existing ones. A case in point is that of the way in which the computerisesd storage and retrieval of records has enhanced some aspects of detective work without significantly altering the overall nature of this work. This is not to argue, however, that detective work should not be changed, but the case for this, while it might involve computerisation, is not solely, or even mainly, a technological matter.

Further, it is important to resist the urge to generalise from, say, detective work to other aspects of police work or to other types of work where very different

1 This is an aspect of the regular call, again in the UK context, for the establishment of a national police force. So far these have been successfully resisted, though the size of local constabularies has been growing over the years through a succession of amalgamations, and IT technology is becoming increasingly used as a means of facilitating inter-constabulary cooperation.

considerations might well apply. In many respects, the issue is about the kind of service the SOS can and could meet and perhaps this needs to be determined initially. In other words, though this is not to minimise in any sense these challenges, the issue is not so much technological but one of design options given the kind of service the SOS would need to meet

What follows is an attempt to specify some of the services an SOS would need to offer to allow for the flexible delivery of these services, ensuring for their 'trustability', and the adequate support of a variety of work activities within a variety of settings.

9.5.6 A Shared Object Service: Some Suggested Properties

One of the major problems in designing such a device for CSCW applications is 'representing' in electronic form some of the subtleties of document use which 'belong' to current manual systems; subtleties which are often 'invisible' because of their very familiarity of such systems. However, it is not the case that all the subtleties need to be 'represented' in the system, either because human beings are able to do such things much better or because they cannot be. Nevertheless, there are properties of manual systems which are likely to become properties of computer record systems, though the latter my perform these services much better. In what follows we try to specify some of these properties on the assumption that such an SOS would be adaptable to a variety of work settings in which records figure largely.

We focus on the following headings:

- topic
- organisation and connectness
- access
- relationship to organisational action
- authentication
- trustability

As before, it is to be noted that many of facets of these issues are closely interrelated.

To some extent, they can be roughly mapped onto some of the issues identified in COMIC Lancs-4-2, 'COMICal Issues on A Shared Object Service' which were set out as follows:

- *Object Representation:*

  This included schemas, shared schemas and metaschemas and, presumably in part, relates to the question of topic.
- *Representing People, Groups and Projects*

  Here the dynamic nature of groups is stressed as an objective of the SOS as well as issues to do with awareness of sharing, locking and history. This relates to issues identified above as concerned with organisation and connectedness as well as access, authentication and the relationship of records to organisational actions and activities..

- *Object Interaction*

  This is concerned with the presentation of 'shared objects' to users and other important interface issues. It also includes reference to interaction paths and the interactional properties of shared objects. Most of the issues as set out above are related to this, particularly organisation and connectedness, access, and relationship to action.

It should be noted that the above mapping is rough and only suggestive. Also, it is important not to treat these as discrete problems except for the practical purposes of discussion and the highlighting of issues.

Topic

This issue addresses the very important matter of what records should be about. We said earlier that records are constitutive of 'organisational objects'; that is, about matters of concern to the particular type of organisation it is. This, of course, says nothing about the details of what these might be, nor is it clear that they could be the concerns of the SOS, though two related aspects stand out:

- formatting
- continuity

Formatting

This follows the usage in 'Paperwork' and refers to the 'stylistic structure' of the document. Earlier we distinguished between relatively 'loosely' and relatively 'tightly' structured documents. The prime example of the latter is the form and would also include many of the documents used in police information systems, such as Crime Recording and the Incident Log. In these cases information is unput according to sequence of specified fields. It not only makes inputting information easier over the telephone, it also makes use of what is already reasonably standardised information and stores it much more rapidly than the manual system was able to achieve. The same fields can also be used for information retrieval. It is this facility which, in part, is responsible for enhancing the use of records among police officers in the constabulary studied.

On the other hand, social work records tend to be 'loosely' formatted in that although there is the usual 'face sheet' information to enter, such as addresses, names, ages, relationships, and so on, much of what else enters the file consists of reports of varying length which do not follow much of a strict format of the type typical of police incident reports. Medical records often contain both types of format.[1]

The point about formatting and its relationship to the topic of the record is that, as in the case of the police, where records are entered by others in response to information received by an officer in the field, then 'tightly' formatted information

1 The distinction is nothing to do with whether or not the document requires interpretation.

has some advantages. However, it is doubtful whether such formats are suitable for all types of records. Social work records would be an example. Accordingly, search facilities when using records aggregatively would need to be devised to handle both types of record.

An issue related to this is the extent to which the SOS would need to handle inter-organisational information as well as intra-organisational. In the case of the police and the legal system one can see that while there are transformational processes at work in the movement of records from one to the other, there is an argument to suggest that the 'basic' information required by both is the same. In any event, even were this not the case, the facility to transfer and move information between organisations might well be an important one and thus have implications for data transfer procedures.

Continuity

This is not necessarily a feature of the SOS but is an acknowledgement that any implementation of such a device in a real world setting would, likely as not, for some time have to sit alongside a manual system. Transferring manual files which are still 'live' in some sense is likely to be a prodigious task. In addition, and for a number of reasons, it may be that in some settings - and legal settings immediately come to mind - authenticity would require original manual documents as the record of authorisation.

Thus, the SOS would have to develop facilities for the smooth interleaving of both systems.

Organisation and Relatedness to Action and Activities

This refers to the ways in which single items are records are grouped, coordinated, connected - in a word, organised - within the work setting. A key part of this is the ways in which records not only initiate action but also attest to the completion of actions.

In work settings the organisation of records tends to be 'congruent' to the organisation of work activities in that the 'who' and the 'what' of a record is very often tied to organisational identifications. This also has implications for the actions of organisational others. Records, and this is to repeat again an important point, are not unsituated, decontextualised features of organisational life but integral to it. The ecology of record use, face sheet formatting, signatures, status identifications, distribution, etc., are all features which, in a manual system, support, reinforce, reflect, in prosaic and everyday ways, the organisational and normative voice that records have. This is, of course, strongly related to issues of authentication.

An SOS would, accordingly, have to contain, for a particular organisation, some 'channelling device' congruent to the organisation's information and document flows. Part of this would, presumably, be a representation within the SOS of organisational identifications and, as important, tied to those identifications some specification of the rights, obligations, responsibilities, entitlements, etc., relevant to the identification-holder's role with the organisation. Such identifications can, of course, be collective.

The above, of course, reflects what is a traditional view of the formal structure of an organisation and would bear much of the same problems and complaints which have been levelled at such a picture, briefly, inflexibility, incompleteness, inertia, etc..

The design choice here is a difficult one. An SOS which simply incorporated the kind of organisation picture sketched above would be a limited one for handling primarily official documents and channels of information and, accordingly, would acknowledge that 'informal' channels are not part of its design remit. These could be left to other 'informal channels' such as personal contact, telephone, email, gossip, and so on. There is a plausible case to make for such a system not least because it reflects reasonably well aspects of organisational life. There are hierarchies within organisations, there are official channels of communication, there are organisational labels for persons, and so on. This is not *all* there is to organisations, and not all organisations display the same patternings of these features, but they are significant features of many organisations and, as such, a fact of organisational life and, through this, a fact of CSCW life.

For example, the Crime Recording Bureau used by the police is essentially a very large data base which replicates and enhances the manual system, as is PNC. They fit into the work activities of the police satisfactorily insofar as that work consists in processing records and accessing well-defined and limited information. This is not, however, all that police work consists in. Some of that work can be decribed as 'intelligence gathering' and can include gossip heard about potentially criminal activities, as well as deliberate surveillance of suspected criminals. Such information is not easily codified, is intimately a matter of interpretation, 'gut feeling', suspicion, hearsay, and so on. Little of this kind of information would stand any kind of legal test, and yet it is an essential aspect to much of police work. It tends to get propogated 'by word of mouth', personal contacts within the force, though some of it does become available to officially established task forces. What is not always clear about such information is who can make 'best' use of it. Who should it be disseminated to. Moreover, while there are current efforts to build intelligence systems by the police, it is not clear how effective they are.[1]

However, it is one of the principles of CSCW - though perhaps one that needs more scrutiny - that groups are dynamic, in the way that organisational structures allegedly are not, and an SOS would need to reflect this. What is not clear is how this principle can be instantiated in particular cases or, indeed, whether it ought to be. Moreover, there is the question of who would initiate any changes within groups, etc. There is no doubt that 'informal' groups develop within organisations, groups which interweave in complex ways within the formal structure - to stick for a moment with this picture. Whether or not such groups should be supported is not a technical question. As the evolution of the network demonstrates, once such a

1 Not least because they come under the Data Protection Act.

system is in place it can evolve and develop in ways that provide access to formal and informal ensembles. As we say, whether the SOS will be allowed to support such activities is a matter of organisational politics.

Perhaps of more significance is the relationship of records to actions 'to be done' and which 'have been done'. Sometimes these will be an instrinsic part of the record in that an action itself produces a record. But this is not always the case. Thus, an SOS would need to reflect the initiation of actions and their completion and develop devices for recording these. Such devices may include active diaries, flags attached to records, queries, etc.. More difficult are those cases where the outcome is a matter of collaborative judgement, as in the case meetings in social work. Such judgements would need to enter the relevant records as well as indicate respective responsibilities and actions.

Access and Awareness

These issues are, of course, closely related to the former but are dealt with separately for reasons of emphasis. They are also issues to do with the security and the privacy of records.

The point has been made that organisations are, among other things, about a distribution of rights and responsibilities and this includes access to recorded information. And, as we have previously discussed, access is not simply about who should know what information the record contains, but also who has the right to act upon, disclose, use that information, that record, in the course of their work. Thus, and for example, civilian inputters know what the Incident Log and the Crime Reporting Bureau contain, but they have no rights to act on that information in the way that police officers have. The same point can be made with respect to secretaries in organisations and many other who produce, organise and manage records, often in routine ways, in what can be described as a *librarian function*. However, it is worth pointing out that those who serve this function are very often essential to providing those who have access to that access in their knowing of where a file, a record, is or where is can be found.

Perhaps more significantly is the relationship of access to established organisational relationships which can be hierarchical, to do with the distribution of responsibilities, and more. It is, however, difficult to say anything more specific about these in the absence of detailed investigations of what they would imply for a 'real world' SOS. We can say, however, that an SOS would need to provide adequate means to service an access function which would specify, for each class of records, who is entitled to see and use them.

The difficult problems arise in designing a suitably graded means of reflecting access rather than an all or nothing matter bearing in mind that 'graded' here does not imply that there is only one dimension to use to determine these matters. What we have in mind here are the uses of records which are for 'informative purposes' rather than directly relation to some action. Thus, a social worker is able, and entitled, to 'go through' the records motivated by, for example, 'seeing what was done in previous cases', 'checking something out', 'finding an address', etc.; in a word, browsing. Such a social worker may browse the records but is not thereby entitled to add to records that are not his/her responsibility.

Supervisory issues are also relevant here. In social work, for example, as in the police and in many other organisations, records are part of a supervisory relationship in which the record, along with other records, are accessible to someone with the responsibility for the performance of others.[1] This can be an activity which is regularly scheduled or one which is initiated by either party or even one which is relatively cursory.

Such subtleties are part and parcel of the cultural 'know how' of organisational members. As far as the SOS is concerned they raise issues of traceability, awareness and versioning which are, of course, relevant in other ways.

Authentication

As we have already indicated there are many ways in which record authentication is established and supported within manual systems. Many of these are conventional including the use of signatures, insignia, original versions, and so on; devices which raise some obvious problems in computerised systems which can produce an infinite number of versions of a document or a record each indistinguishable from the original.[2] A standard way of coping with this problem is to retain the paper record from in a parallel manual system. However, an SOS would have to develop authenticating devices for other kinds of records. For example, a shipping order for a product has to be, for the receiving department, an authentic record of a legitimate instruction. In other words, records which have what we earlier referred to as procedural implicativeness in that actions follow from them, would need some means of authenticating the legitimacy on that issuance.

In some respects, authentication is a matter of custom and practice in that it is relatively rarely the case that organisational members constantly and unremittingly make inquiries into the authenticity of records. Their orientation to these kind of issues are very practical ones in which such inquiries tend to be initiated only when something 'doesn't feel right'[3] Nevertheless, authentication is an issue for the SOS.

Trustability

This last issue is, of course, extremely wide and is included mainly as a reminder that systems have to be trustable. As with a number of other matters we have already discussed, trustability is not any one thing but is about the 'easiness' with which tools are able to be melded into work activities. While this is clearly a matter for informed design, it is also a matter of familiarity, training and the processes by

---

1 This is, of course, a task done with variable degrees of scruple and thoroughness.

2 In some respects the advent of high quality photocopiers raises this issue with respect of paper based systems.

3 Other cases are where very poor systems tend to get subverted or 'got around' in order to 'get the job done'.

which tools, of whatever kind, become 'enculturated', so to speak, within the work setting.

It is important here, and to repeat the point, not to think that any one thing underpins trustability in a system. And it is not necessarily about building a system that does not fail regularly. It is more to do with 'getting to know' what the system will do, 'getting the system to do what I want it to do' within the kind of work that is done. This is, once again, an area where much more work needs to be done.

9.6 Some Short Concluding Remarks

This working paper has attempted a general review of issue in the social organisation of document use, using for illustrative purposes material drawn from the work enviroments of the police, and social work.

What is has tried to draw out are some features of document use relevant to the design of an SOS for CSCW applications. It has highlighted the importance of making clear the assumptions on which such a design would be conceived and also made some suggestions as to the kind of functions it would need to incorporate to retain some of the features of what we called manual systems; features which have much to do with the socially organised features of work within organisational settings.

It is also clear that much work needs to follow from this and other working papers which contribute to the research in Strand 4. In particular, more research needs to be done on the details of record use within organisations as well as some effort to inform the SOS model, even if only in general terms at this stage, what kinds of services it should provide bearing in mind that such services would need to be 'organisationally adaptive and adaptable. In this regard, it might be useful to envisage 'what if' scenarious based on studies of work activities, to attempt to identify some of the consequences of incorporating CSCW applications with respect to record use.

While there is, within thecomputer science research community, now a familarity with electronic systems as indispensible tools for computer science, so much so that they have developed their own etiquette and understandings of use, there is little doubt that incorporating such systems into much of organisational life is a much more arduous task. Such acceptance is likely to be a slow business though more effective to the extent to which systems can capture some of the subtleties of manual systems. Capturing such subtleties is not only a matter of interface design though this is an important matter that will need to be addressed, particularly in the direction of creating effective and smooth access to document stores. It is also a matter of the appropriate infrastructure for distributed and interactive systems. And, as mentioned a moment ago, informing the design of such subtleties will require more detailed studies within various domains.

There are other matters which are of relevance here and which need to be looked at. We particularly have in mind issues to do with what is often referred to as organisational memory as well as the vital matter of organising and supporting the retrieval of information. One of the things an electronic world is good at is storing vast amounts of information but in the context of human affairs the amount of information is not so much the problem; organising it appropriately is which is another important aspect which the design of the SOS must address.

10. Requirements for the COMIC Shared Object Service

Tom Rodden and John Mariani
University of Lancaster

Björn Eiderbäck and Per Hägglund
KTH

Yngve Sundblad (KTH)This chapter is a set of requirements for the COMIC shared object service. A list of services of particular importance for CSCW systems and applications have been formulated, and, in addition to some basic underlying concepts, each service is described. The document also sets out the realtionship between the shared object service and other services provided to support cooperative work.

10.1. Introduction

Shared information plays a central role in many CSCW systems. It is often the primary means used to develop a shared understanding across a number of user working within the systems. The nature of this sharing and the forms of cooperation it facilitates vary greatly from system to system. For example, systems such as Groupdesk (Greenberg 1991) allow a number of potentially remote users to work together in real time while annotation systems such as QUILT (Fish 1988) allow users to work in a time-independent manner.

This chapter briefly examines the nature of shared information within cooperative systems before outlining the requirements for a shared object service to be developed as part of the COMIC project. The aim of the shared object service is to provide an appropriate set of storage mechanisms to allow information to be shared in a manner which encourages rather than prohibits cooperation.

Despite an acceptance of information sharing as one of the fundamentals of cooperative working most CSCW projects have designed and implemented a data model that they believe best meets their aims. In so doing, they have failed to capitalise on any of the facilities provided by database management systems and the potential benefits of standardisation for sharing across applications (Mariani 1991).

Most CSCW systems are in some agreement on the nature of the information they use and have settled on some form of object model. As the provision of distributed object systems continues to gain speed in both the commercial and research fields, we must ensure that these systems meet the requirements of CSCW. It is the intention of the COMIC shared object service to examine the nature of these requirements and provide appropriate cooperative sharing techniques. However, before considering the nature of a shared object service in any detail it is worth the context within which a COMIC shared object service must exist.

10.1.1. Sharing within CSCW

Cooperative applications provide the sharing of information at a number of different levels of abstraction each providing different facilities. This sharing is provided at three significant levels of abstraction:-

The windowing level

A number of CSCW applications have focused on the provision of shared screens which provide real time communication between remote users at the windowing level.

The presentation level

A number of application development environments have focused on the used of presentations of shared objects to provided appropriate facilities to support cooperative work.

The object level

CSCW systems have considered the use of shared information as a means of mediating cooperation. The most notable example of been in the use of multi-user hypertext systems.

Shared Window Systems

Shared windowing systems have dominated much of the initial considerations of CSCW systems and highlight the ability to combine both computational and communication facilities within cooperative systems. The most notable example of shared interfaces is the development of WYSIWIS within the CoLab project (Stefik et al 1987). Lauwers and Lantz (1990) outline two approaches for the development of multi-user interfaces. The development of special purpose applications which are *collaboration aware*, and the adapting of existing single user applications to provide *collaboration transparent* shared applications. Examples of work which exploit collaboration transparency include Vconf (Lantz, 1986), Rapport (Ahuja, 1988), SharedX (Gust 1988), and MMConf (Crowley 1990).

Most of the collaboration transparent synchronous interfaces exploit network based window systems such as X or NeWS. These systems assume a client-server or virtual terminal model of interaction where the application and the display are separate and connected by an interaction protocol. This approach allows local graphical events to be handled by each display with interaction details being passed to the application using this protocol. A significant problem with this approach is that an assumed model of cooperation based on many readers but only one writer is embedded within the system. As a result it is difficult to customise these systems and any tailorability which exists has concentrated on the development of techniques to support different forms of turn taking protocols (Greenberg 1992).

Shared presentations

In contrast to the transparent approach outlined above, collaboration aware solutions provide facilities to explicitly manage the sharing of displays between different users. This approach has been adopted by a range of applications including Cognoter (Stefik et al 1987), Grove (Ellis 1988) and rIBIS (Rein 1991). These multi-user cooperative applications focus on the *presentation level* and can support a number of alternative user interfaces and many of them provide facilities which allow individual views to be supported. A number of systems have examined how this arrangement can be generalised to provide supporting architectures for a variety of applications. This approach is adopted by both the MEAD system(Bentley 1992) and the Rendezvous system (Patterson 1990) which has the ability to support different user views of application objects and interactions with these views. These systems maintain a clear separation between *underlying objects*, which contain the application semantics and *interaction or presentation objects*, which maintain details of users' views on these objects.

A second example of presentation or view sharing is offered by the on-going research on *shared spaces*(for example, Tang 1990, Bly 1990, Greenberg 1991 and Ishii 1991). The approach adopted here focuses on the capture and visual representation of shared information (most often in the form of drawings). These systems provide a space upon which users can collectively express and structure their ideas in a relatively unconstrained way. These are most often used by a number of designers to represented shared designs. The majority of systems have concentrated on the creation of new information and the use of "real-time" drawing as the primary means of expression. Many of these systems have been directly informed through an observation of their use (Tang 1991).

Shared Information systems

A number of different CSCW systems exploit the use of a shared store to support cooperative work. This form of cooperative system is both time and location independent and users interact through some form of shared information space. This model of interaction is often augmented by direct user communication. This is normally provided by either electronic message systems or by the use of an audio or video connections. This model of cooperation has been exploited within a wide range of cooperative systems.

The most notable of these have derived from multi-user hypertext systems such as KMS (Yoder 1989) or SEPIA(Haake 1992). These systems rely on the ability to represent information as discrete objects which can be linked together to form network structures. This has been most notable within argumentation and rationale systems such as SIBYL(Lee 1990) and rIBIS(Rein 1991). However, shared hypertext facilities are also exploited within cooperative authoring systems such as Quilt

(Fish 1988) and Coauthor(Hahn 1991), comment and review systems such as ICICLE(Brothers 1990) and in software maintenance(Lougher 1993).

In addition, to the facilities provided by these multi-user hypertext systems a number of cooperative application make considerable use of shared information repositories. These information stores can be used to store a range of information such as directory knowledge within X500 and organisational information (Ackerman 1990, Prinz 1993)which can be readily browsed by a number of cooperative users. This is reflected in a growing interest in concepts of organisational memory or record within CSCW (Conklin 1991, Berlin 1992).

The intent of a shared object service is to provide one section of a necessary platform for the development of cooperative systems. Thus rather than consider the problem of CSCW as been essentially a user interface issue we are examining the more fundamental information storage implications which emerge from a consideration of cooperative work. In order to this within COMIC we need to make a clear separation between the facilities to be required for shared objects and those which need to be provided to support 'real-time' shared interface facilities. This separation is realised within COMIC by the provision of closely related services which manage a particular set of facilities. Two services of particular note are the shared interface service and the associated shared object service. The shared interface service may often make considerable use of the facilities provided by the shared object service. The functional relationship between these services and the user applications which make use of them are shown in figure 10.1. The diagram is intended only to illustrate the functional relationships within the service rather than to suggest any particular architectural structure or seperation for the service.

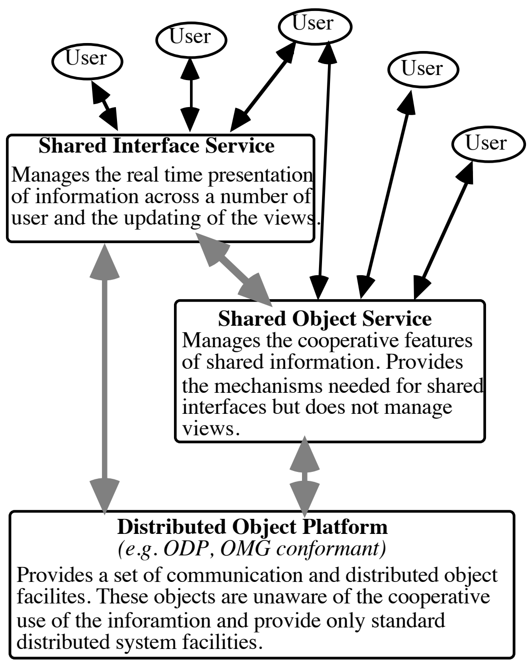


Figure 10.1: Relationships between the shared interface and object services

This document is particularly concerned with the services to be provided by a shared object service. Related facilities such as those provided by the shared interface service are discussed in separate documents. The separation between the shared interface service and the shared object service allows us to highlight the close relationship between a shared user interface and the form of cooperative support provided. For example, one form of shared interface service could be workstation based visualisation facilities such as those provided by MEAD system (Bentley 1992). Other alternatives include the shared interaction provided by the knowledge net (Marmolin 1991) or the spatial representation and interaction provided by the COMIC spatial model (Benford 1993). The point here is not that any of these are particularly appropriate but that all can make use of some form of shared object service. The provision of a service thus provides us with both a stable platform from which to consider supporting cooperation through shared objects and a means of integrating a number of approaches to cooperation in a seamless manner.

10.1.2. The role of the service

The explicit aim of the COMIC shared object service is to provide a set of services which allow objects to be shared by a community of users. The distinguishing feature from existing multi user storage services is the focus on sharing across a group of users and the provision of mechanisms which support the management of this sharing. It is intended that the shared object service provides a set of facilities for a group of users which abstracts from the properties of underlying infrastructure to provide a well defined abstract set of services. This offers the advantages of portability and allows the project to consider the services which need to be provided in terms of a abstract computational model which can then be realised on a number of different distributed object platforms. The overall structure of the shared object service as an abstract veneer which sits on top of existing distributed infrastructures is shown below in figure 10.2.

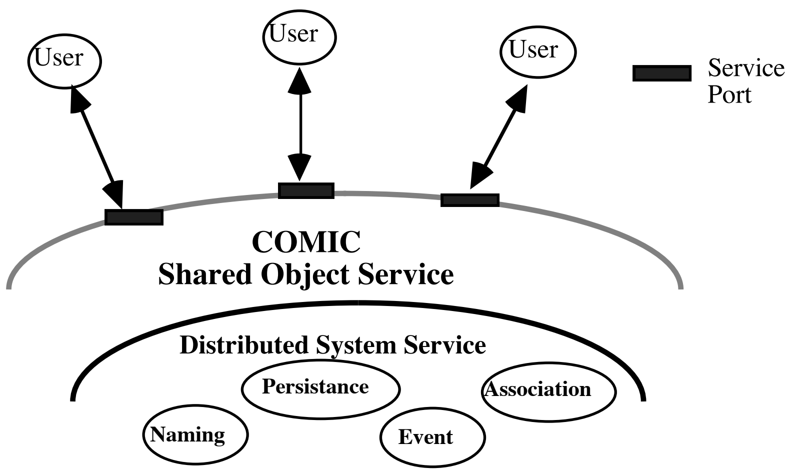


Figure 10.2: An overview of the shared object service

The shared object service is controlled through a set of publicised commands issued to a known service port which allows users and applications to specify the facilities they require. This document is intended to describe the requirements for the central features of this service rather than to consider the exact means by which these facilities are presented to users,

Objects within the service

The shared object service provides an abstraction over existing distributed object services. Thus rather than focus on the development of an appropriate object model for cooperative work the shared object service aims to complement existing work on the development of object models by focusing on the cooperative sharing of information. To both avoid any unnecessary duplication of work and to encourage as wide a dissemination as possible of the object service work the service will focus on augmenting the facilities provided by existing distributed object systems and services.

To allow this to occur it is intended that the SOS object model will follow the work of the OMG efforts . We believe that for basic and general object services, the work described in documents such as the OMGs Joint Object Support Service will provide adequate functionality. It is not the purpose of the COMIC project to redefine or reinvent these kinds of object services, but rather to concentrate on those aspects and services of particular relevance to CSCW systems and applications, some of which may be in direct opposition to conventional wisdom and technologies. We thus adopt the use of object definitions within the OMG services (OMG 1990).

The OMG object model defines a core set of requirements that must be supported in any system that complies with the Object model standard and provides a useful baseline for the project. The model defines common object semantics for specifying the externally visible characteristics of objects in a standard and implementation-independent way. The externally visible characteristics of objects are described by an interface which consists of operation signatures. Each operation signatures consists of an operation identifier and a parameter specification. Object interfaces embody the external view of both object behaviour and object state.

Objects within the model are grouped into types and individual objects are instances of their respective types. A type determines what operations can be applied to its instances and participate in subtype/supertype relationships. A type is essentially a template describing the common expected semantics of an object. Types have implementations which are typically sets of data structures that constitute stored representations, and sets of methods that provide the code to implement each of the operations whose signature is defined by the type.

Inheritance and *subtyping* (of an object interface or implementation) provides object interfaces with the ability to be dynamic (changing as the system changes) and to present themselves in such a way as to give adjustable and extendible functionality. These models of inheritance and subtyping closely resemble those found in object-oriented programming languages and object-oriented databases.

Within OMG, subtyping is a relationship between types based on their interfaces. It defines the rules by which objects of one type are determined to be acceptable in contexts expected by another type. Inheritance is a mechanism for reuse. It allows a type to be defined in terms of another type. Subtyping requires that for an object to be a subtype of another the name of the operations must match, and the number and types of parameters and results are the same. A subtype can have multiple supertypes.

Building upon existing services

Our expectation is that a shared object service developed as part of COMIC needs to exist along side services defined by other bodies. Thus we expect to make use of services provided by bodies such as the OMG. Consequentially we have chosen to adopt a philosophy of augmentation with the services provided by the shared object service augmenting those provided by the OMG joint object support services. Where appropriate we shall reference the appropriate portions of the OMG services we will build upon.

The OMG joint object service provides a number of different services which a COMIC shared object service can in turn exploit. In particular, a number of significant services are of direct relevance to the COMIC shared object service and are assumed to be provided by a supporting distributed object service. Services identified by the OMG include:-

A Naming Service

The OMG naming service provides support for binding names to objects and supporting and defining a naming context for objects. A shared object service may exploit the naming service provided by an OMG conformant service. However, the shared object service may augment this by providing a cooperative context for the object naming.

An Event Service

The OMG event service provides an event channel to support asynchronous communication between multiple suppliers and multiple consumers. The OMG service also supports event types and data. The base level mechanisms provided by the OMG can be augment with the provision of more structured events and event subscription and propagation tailored services to support cooperative work.

An Association Service

The OMG association service provides mechanisms to allow an object to reference another . The service supports binary relationships between objects. These services can be used directly by the shared object services by been mapped up onto relation objects within the service. The mapping up of the association service to relation objects in the services allows the evolution and expression of cooperatively defined relationships within the shared object service.

A set of life cycle services

OMG provides a set of services which manage the creation, movement, copying and removal of objects. These services are collectively termed the object life-cycle services are intended to support an object throughout all aspects of the object's lifetime. These services will be used directly by a shared object service to manage objects represented within the COMIC shared object service.

Persistence storage service

The OMG services provide a mechanism to support the persistence of a shared object service. The service is defined in order to support persistence extensions to programming languages. This service may be exploited to ensure persistence within the COMIC shared object service. However, persistence could equally be supported by direct use of some for of object store or database.

10.2. Aspects of the Service

The rest of this document considers the different functionality required of the shared object service. The focus of this examination in is identifying how various aspects of the service may be realised as part of the service. To aid this the functionality needed from the shared object service is considered by examining different elements of the service. The initial set of service elements considered in this document are outlined in the following table with a brief description of the particular responsibilities of each service element.

10.2.1. Architectural Implications

The shared object service while presenting a unified interface can be constructed from a number of different components. Each of these different components can in turn characterise the properties of the service they provide. Different parts of the shared object service may provided in one of a number of ways. Service components can be provided as:-

- A *kernel feature* of the architecture. These features may be provided as a set of system calls and undertake a particular task on demand from a client. Service requests will normally be from either an application or a browser acting on behalf of a user. It is unlikely that these facilities will be directly accessed by users. Kernal features are used to provide tkey persistance features of the service.
- An *abstract manager* within the service can provide additional features by exploiting underlying facilities. The abstract managers can act autonomously exhibiting a range of behaviour in order to provide a particular service.
- A *service agent* gathers together a set of kernel features and abstract managers to provide a particular form of object access. It is expected that service agents will be directly accessible by users and will provide a user interface to either present particular information or to allow particular properties of the service

Service Element	Description	Section
Locking	A set of facilities to support coordinating concurrent access to shared objects by cooperating users .	Section 5
Versioning	Facilities to support and manage the recording and maintenance of versions in cooperative setting.	Section 6
History	A set of history mechanisms which promote asynchronous cooperation by recording the history of object use.	Section 7
Access	Facilities which control users ability to access different aspects of the service.	Section 8
Queries and Views	Facilities to allow the definition of different views of objects within the service and the use of these	Section 9
Events, Filtering and Subscription	A set of service facilities which allow the handling and management of events within the service, and which allow different foci of interests to be defined for different groups of users.	Section 10
Awareness	A set of facilities which allow objects to manage the level of awareness they provide to different users of the service.	Section 11
Interaction and Presentation in co-operative applications	The facilities provided to support different users interaction with objects in the shared object service. This includes which methods an object presents and how these are invoked. The presentation of different objects and aspects of the service to end users. This is achieved through a specialised user agent which determines a set of appropriate presentations for each user.	Section 12
External Refs.	Facilities provided by the shared object service to allow some form of bridging between electronic objects within the service and other information objects outside the service	Section 13

to be altered. The shared interface service can be considered as one form of service agent.

The different aspects provided by the COMIC shared object service have a number of architectural implications for the provision of the service. A number of distinctions can be made as to which of the various aspects of the service are central service features while others are provided as additional service. For example, storage and representation facilities can be central kernel facilities within the service while external reference features are provided as additional services. A more significant division is between the shared storage facilities provided by the service and the shared representation facilities which have been delegated to the shared interface service.

The following tables summarises the different aspects of the service identified to date and the means by which the will be provided within the service.

Service Element	Type of component
Object Representation	Supporting distributed object system
Shared Object Locking	Kernel Feature of the system
Object Versioning	Kernel Feature of the system
Shared Object History	Kernel Feature of the system
Access Control	Abstract Service Manager
Views	Abstract Service Manager
Events	Abstract Service Manager
Filtering and Registration	A Service Agent
Awareness	A Service Agent
Object Interaction and Invocation	Abstract Service Manager
Object Presentation	A Service Agent
External Refs.	Abstract Service Manager

10.3. Object Interaction and Interfaces

The COMIC object model is influenced by existing object models provided by Smalltalk (Smalltalk 1992) and popular distributed systems such as ANSA(1989) based upon ODP (ISO 1991)), and CORBA(1991) (based upon (OMG 1991)). Within these models the concept of *object interfaces* plays a central role in defining the behavioural semantics of objects. In Smalltalk everything is considered as objects and interaction between objects are alternatively done through direct messages passing or via some level of indirection, such as dependency relations and active values. The ODP and OMG models adopt a client-server philosophy with client objects requesting services from server objects using some form of message passing or remote invocation. In general when a client wishes to access the facilities provided by a server, it obtains a reference to the service interface it requires from the platform. It then “binds” to this interface and can subsequently issue calls to the bound interface which are handled by the server and normally return the result of the call to the client.

Within the COMIC shared object service each object provides a basic interface which allows its state to be inspected and manipulated using simple put and get attribute methods. The put and get interface provided by the shared object service allows access to a basic interface to shared objects. Additional interfaces can be added to this basic interface. Each interface contains a list of access points for a set of procedures which cause the shared object to exhibit particular behaviour. The addition of interfaces to the shared object over rides the basic object manipulation interface to prevent comprising the state of any shared object. The shared object service needs to provide facilities which allow these interfaces to be both represented and shared.

10.3.1. Interaction with the shared object service

Objects within the shared object service are represented as abstract units which embody both state and procedure. Objects which publish an interface which make the services the provide available to other objects in the service. A important feature of this service is the means by which users interact with both objects and the facilities provided as part of the shared object service.

There are a number of ways of the functionality provided by a shared object service could be realised and presented. Consider as an example locking which is a fundamental component of any multi-user system . This can be provided in one of a number of ways:

As a basic mechanism of the service

Traditionally, within database management systems, this would certainly be implemented as a fundamental component of the DBMS service.

As part of each object

Within the object-oriented database world, this functionality might be implemented as part of the methods and state associated with the most basic object type. Thereafter, the user can safely assume that every object type possesses the ability to lock access to (portions of) its state.

Using multiple inheritance

A further approach might be the use of an associated object linked using multiple inheritance to the object being locked. For example, we may have a "person" object and a "lock" object. By creating a new shared subtype, we can create a "lockable person" type.

Using adapters

An object-oriented alternative would be to use interface adapters which dynamically handle objects' invocations (see below for a description of adapters).

The philosophy adopted by the shared object service is to provide all aspects of the as objects which can be interacted with using a similar set of mechanisms. Thus both objects within the services and the facilities provided by the service can be used by clients in an orthogonal manner. Given this orthogonality it is important that we maintain a separation between the behavioural semantics of the objects within the service and the management facilities provided by the service. For example, the use of multiple inheritance in this case of locking mixes together management and behaviour semantics and potentially overloads the class hierarchy.

We believe that it is vital to separate the behaviour of objects from the management of objects. We propose to use interface adapters which dynamically handle object invocation to support this separation in a clean fashion. This model has been popular for many years in distributed systems. Perhaps the most extreme examples is the use of delegation, which passes messages onto other objects in order for the service to be provided (delegation) or the use of reflection that dynamically allows for changeing of objects' behaviour without affecting the original objects' descriptions. We would also argued that inheritance is an inherently centralised data structure which can be difficult to distributed appropriately and some form of interface adaption is more readily suited to distribution.

Two concepts are important in considering interaction with an object in the COMIC shared object service; interfaces and interface adapters. Interfaces describe the means by which a client interacts with the services provided by the object. Interface adapters provide facilities which abstract over these service to provide different services to users based upon context. The service provides mechanisms based to create and manipulate both interface and interface adapters.

This means that the COMIC shared object service presents two different means of interaction at different levels of abstraction for the objects within the service. An application or user can interact with an object by:-

- Access to the basic interface provided by the object.
- Access through an interface adapter which abstracts away from the object.

Currently, the relationship between these different forms of interaction is one of abstraction. Thus an interface represents an object and an interface adapter ID replaces the ID for an interface it represents.

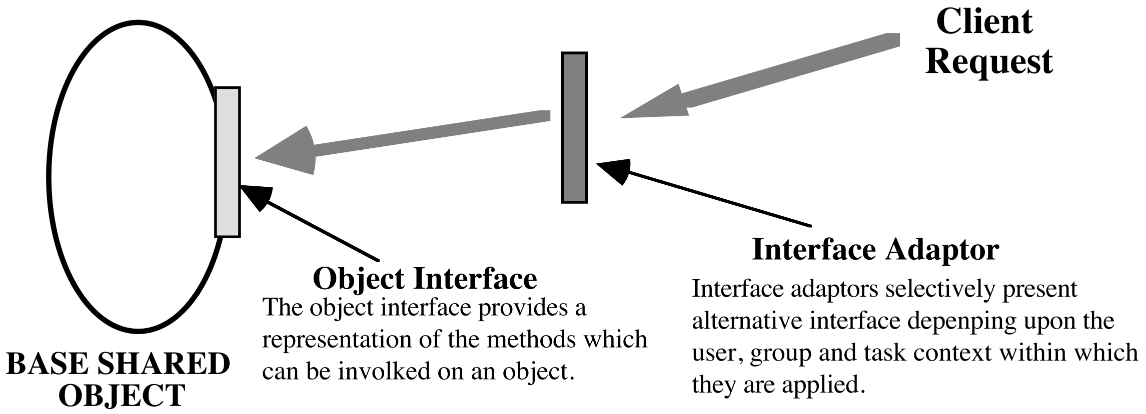


Figure 10.3: Interface, Object Use and Interface adapters

10.3.2. Interfaces

Object interfaces within the COMIC shared object service are analogous to the use of interfaces within distributed platforms such as ODP[1] and ANSA. We can assume that interface details are described using some form of Interface Definition Language such as that suggested by either ODP or OMG. We would expect this definition language to have strong similarities to those provided by most distributed systems. We do not see the COMIC shared object service becoming overly con-

[1] The exact set of mechanisms provided within ODP for specifying interfaces is currently still being defined but the principles appear to be reasonably fixed.

cerned with the exact details of how methods and parameter details are defined and leave this as an issue for supporting platforms. For the purposes of the shared object service a sufficient overview of an interface is to consider it as an access list. Thus each node requiring services from SOS contains information about used methods invocation protocols, parameters and the methods location. This overview of interface within the shared object service is shown in figure 10.3

Interface Adapters

Object interfaces can be adapted to provide contextually sensitive interaction with the shared object using object adapters. Object adapters provide a level of indirection between object interfaces and user of objects. This level of indirection allows interface adapters to provide additional facilities to manage the invocation of methods depending upon the context within which they are defined. Thus object adapters can be used to support access, coordination of users, and task sensitivity. The basic structure of object adapters is as a set of mappings which map method names from those presented to users to those used within the object interface.

This approach maintains a clear separation between behavioural and management semantics and builds upon the use of adapters in OMG and Smalltalk. Object adapters within OMG clad units of functionality provided by different programs to allow them appear as objects and in Smalltalk they even are first class objects. They also provide a set of management facilities which support the clad object within an OMG platform. The shared object service builds upon this separation to allow us to focus on the cooperative sharing of objects. We provide a set of interface adapter objects which provide facilities to manage and support the cooperative features of shared objects. The use of adapters in this way provides a number of benefits.

- A clear separation between the behavioural semantics of objects and the management of objects.
- An explicit identification of the cooperative aspects of shared objects
- The ability to experiment in order to investigate the correct requirements for object sharing in a cooperative setting
- A simple mechanisms which allows a the shared object platform to evolve to support new functionality as it is identified.

If an object requires locking services, an interface adapter is placed in front of the object itself. Requests that were originally targeted to the underlying object are now directed to the adapter. The adapter contains “state” information relative to the locking functionality it provides. Moreover, it contains the “algorithm” or “protocol” used by that particular locking regime. As traditional locking strategies have long been recognised as unsuitable for CSCW applications (Grief 1987, Rodden 1992, Barghouti 1991), by instead embodying the locking mechanism(s) in a set of adapters, this allows us the flexibility to prototype and experiment with a set of mechanisms.

10.4. Information Modelling Facilities

This section considers the facilities provided by the shared object service to allow the representation of different kinds of object and their relationship to each other. The set of mechanisms provided by the service allow different object structures to be represented and modelled.

An analysis of existing information models as used by CSCW systems and applications allows us to form a synthesis of object data types that are potentially useful across a wide spectrum of situations. It is possible to identify three core object types which are generally useful (in fact, these are so basic that they find application in general data modelling with objects) : the single object, a collection of objects, and objects which represent relationships.

10.4.1. Aims and Scope

The use of the object service to model different classes of information is based on the use of a small set of system object types defined as part of the object service. Within the service there are principally three different basic types of objects : a simple object, a relationship object and collections which act as containers for a number of different objects. It is intended that the services be as orthogonal as possible. This implies that operations on simple objects can also be executed on collections of objects; where an object can be a simple object, a collection, a relationship etc. These are not the only "general" object types; for example, as we will see later, we need a query specification : this, however, should only be an instance of the basic object type.

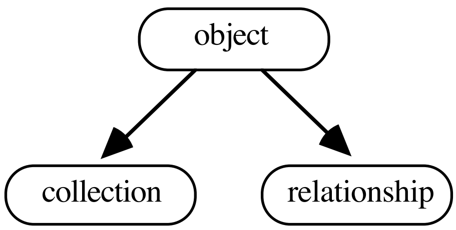


Figure 10.4 : Basic modelling types in the SOS

The shared object service exploits three core forms of object to model information within the service:-

- *Basic objects* with a state and a set of interfaces
- *Relation objects* can be used to form links between objects
- *Collection objects* are also first class objects which can store collections of objects.

The facilities provided by relation objects and collection objects are considered in turn. It should also be borne in mind that to ensure orthogonality wherever a single object appears, a collection may also appear.

Relation Objects

Relationship objects allow objects to be linked together to model that some form of relationship exists between them. Within the COMIC shared object service this is done using a link object. A link object is created by converting any object in the service by the addition of a links field which holds distinguishes it as a link object and holds details of the objects it is linked to. In addition, each object within the service holds details of which relationships it is involved in.

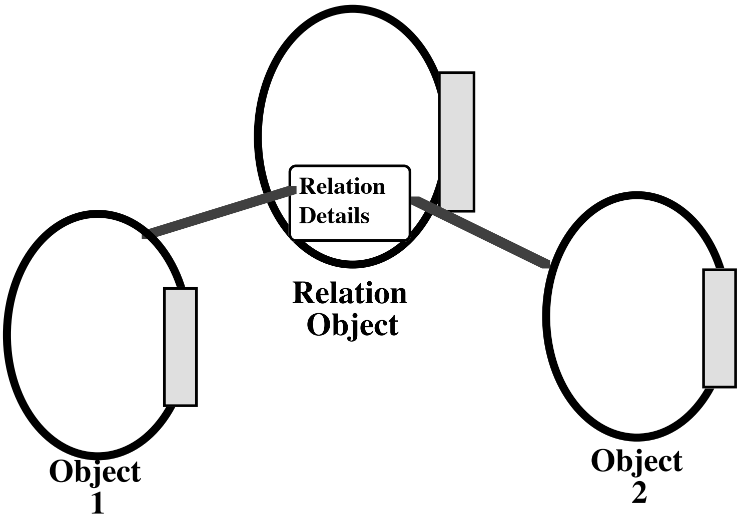


Figure 10.5: A relationship with the COMIC shared object service

Since relations are objects in their own right the details of the relation object associated with the given ID can also be manipulated using the object manipulation interface to alter both the state and interface provided by the relation object.

Collection Objects

Collection objects allow sets of objects to be gathered together under the auspices of a single object. It is possible to think of a collection object being created as the result of a SOS operation (i.e. the result of a query) or being “manually” formed as a user specifically makes an object a member of a collection. Consider for example, the case of an invoice passing through the office described in chapter 8. As a document passes through between users other documents are attached to it and the document soon becomes in itself a set of associated documents. Consequently, collections within the service need to have the properties of first class objects with the ability to act as a single object when necessary.

Once a collection object exists, it is possible to reason about its contents as a single entity; moreover, it should be possible to manipulate the contents as if they were single entity. Within the shared object service two types of operation are handled by collections, *inspection operations* which interrogate object state and *construction operations* which alter object state. Construction operations are handled by passing the invocation onto objects within the collection. For example, if a collection represents a set of employees, it is possible to increase every one's salary by 10% by sending a single message to the collection . While this may is in practice executed by the collection object passing the message onto individual elements the user of the collection perceives this as a single invocation on the collection object. Inspection operations on objects are handled by specialised collection object operations which in turn invoke inspection operations before collating the result for return. These specialised inspection operation may include computational features to compute results derived from the set of objects in the collection.

The use of relationship and collection objects within an object service is not new. There are a number of possible implementation issues but they are not important within the auspices of this document. The OMG association service defined as part of the joint object services is a possible implementation of the relationship objects. The advantage provided by relationship objects is that they allow the semantics of relationships to be directly expressed within the shared object service. Many of these semantics relate to the cooperative use of the objects related together. For example, consider a relationship linking objects within an argumentation network. A particularly link may be hold only for a subset of users within a cooperating group.

10.5. Locking

The locking of information within a multi-user setting has two distinct effects. It ensures the integrity of the information and reduces the possibility of error caused by simultaneous alteration. Locking also provides a means of mediating and controlling the work taking place around the shared information. Potentially, locking provides a means of coordinating access to information, sequencing the use of information and identifying the parties using the information. To investigate the use of locking in a cooperative setting requires us to separate the *locking mechanism* which realises locks upon information from the *locking policies* which control how information is locked. The intent within the shared object service is to maintain this separation and provide a means of making the locking policy explicit.

As an initial proposal for locking, we have chosen to identify a number of policies and lock types along the same lines as those used in object base systems such as ONTOS. For example, ONTOS supports three user lock types : ReadLock, WriteIntentLock, and NoLock. ReadLock indicates read only access; WriteIntent is when the object is intended to be modified. NoLock is a special case where an object can be read but subsequent changes to the object are of no importance to the reader. There is also a WriteLock but this is for the exclusive use of the ONTOS server.

The ONTOS policies are RWConflict (Readers and Writers Conflict) and NoRWConflict. RWConflict is the traditional conservative policy, and NoRWConflict is the optimistic policy where all concurrency checking is deferred to the end of the transaction.

There is also the concept of NotifyLocks. If an application wishes to track changes made to an object by another application, it can set notify locks[1].

One of the key concepts identified thus far in COMIC in the notion of “awareness”. Awareness in the SOS is concerned with users being aware of the presence and activity of other users within the system. Early discussions and analysis of database systems (Mariani and Rodden 1991) and some prototyping work (Mariani and Prinz 1993) seem to indicate two possible natural points in database technology to hook in awareness supporting mechanisms. One is locks and the other events. While current personal preference tends towards the use of events, locks are still a possibility. We would propose that the SOS locking system is extended by the use of “identified locks”. This means when a user (or an application executing on behalf of the user) locks an object, the user’s identification is part of the lock. This gives us the “who” information. In certain applications, it may be useful to record the reason the object has been locked : this gives us the “why” information. This “who/why” information goes a long way towards providing a useful form of awareness.

Locking, transactions and concurrency control are a very complicated part of database technology. Research needs to be carried out into providing a range of lock types and transaction policies; work has to be done investigating how this spectrum of choices can be made to work together in a sensible way.

10.5.1. Supporting multiple policies

One way of providing locking within the shared object service is to have as a central feature of the service a set of specialist interface adapters which provide a set of specific locking policies. Thus the interface adapters would contain a list of users information, a locking list and an appropriate locking policy. Diagrammatically the arrangement would be as shown in figure 10.6:-

1 We believe that subsequently the "monitoring" application has to "poll" the "system" to find out the status of these "notify locked" objects, but as we have little experience of programming with ONTOS we are not sure.

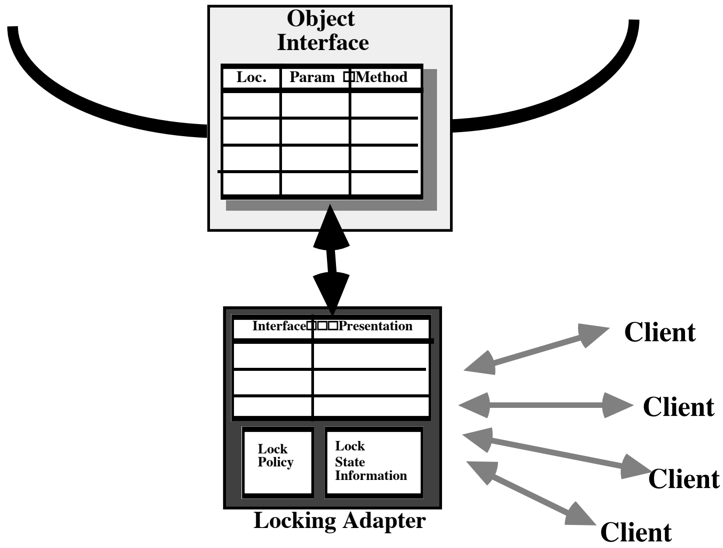


Figure 10.6: Locking Interfaces Adapters

The lockable interface adapters will provide a set of management facilities which will allow the locking policy to be modified. In addition, locking adapters could also allow the representation of aspects of the activities taking place outside the service and from these infer certain locking conditions. For example, a lock adapter could contain the stage of a project and from this stage infer the locking policy which can be applied. We can envisage a locking adapter which provides the functionality of the ONTOS lock scheme, extended with the “who/why” information as the starting point for our investigation.

10.6. Versioning

The shared object service needs to be able to represent different version of objects. What make this particularly interesting within the shared object service is the impact of CSCW on version control techniques. A number of observations can be made about versioning within respect to CSCW that need to be reflected within the shared object service.

- The number of versions will increase dramatically
- The role of a version manager as a single individual may be problematic
- Different version strategies may need to co-exist
- The merging of alternative versions will become significant
- The detachment of users who take copies of particular version will be significant.

Version control strategies within database systems are closely associated to the need to be able to retrace the evolutionary stages in the creation of an item within the database and the relation of different items at a given time. Many of these aspects of versioning may be tackled by careful consideration of the use of the history mechanisms within the service. In addition a version also has an association with the concept of a frozen or agreed example of an item. It is this representation of version as some agreed status of an item that the versioning mechanism will focus on.

In particular we wish to consider a variation from the existing view of versioning and version control systems. The accepted view of version control is that an item is altered during some process and at a given stage in this alteration it is decided that a new version of the item exists. A new version is then created and made public. This current version then provides the basis for an continuing changes to the item. Often a number of alternative changes need to be explored. In this case the version path branches to show alternatives. The traditional versioning process is shown below in figure 10.7.

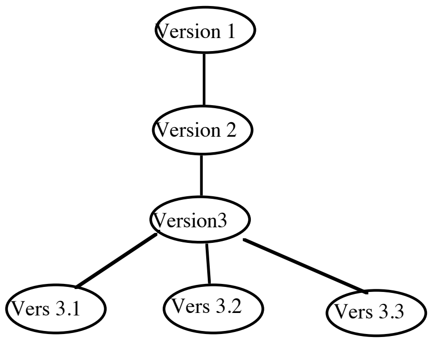


Figure 10.7: The traditional relationship between different versions

This traditional view is enabled by a number of things. A single manager (human or machine) determines when a change is significant enough to constitute a new version and when an alternative allows the version tree to branch. However, the situation within the cooperative setting is a more problematic. Objects are shared by a number of users who each may make alterations to these objects. If we say that each users alterations may yield new versions we have massive branching problems in the tree. Equally well we have significant issues in how these different versions are then merged to yield the next version in the linear tree. In terms of management of the version the significant problem we have to address is the coordination of version generation across a community. For example, if three people can simultaneously generate versions the traditional view show in figure 10.7 becomes somewhat chaotic.

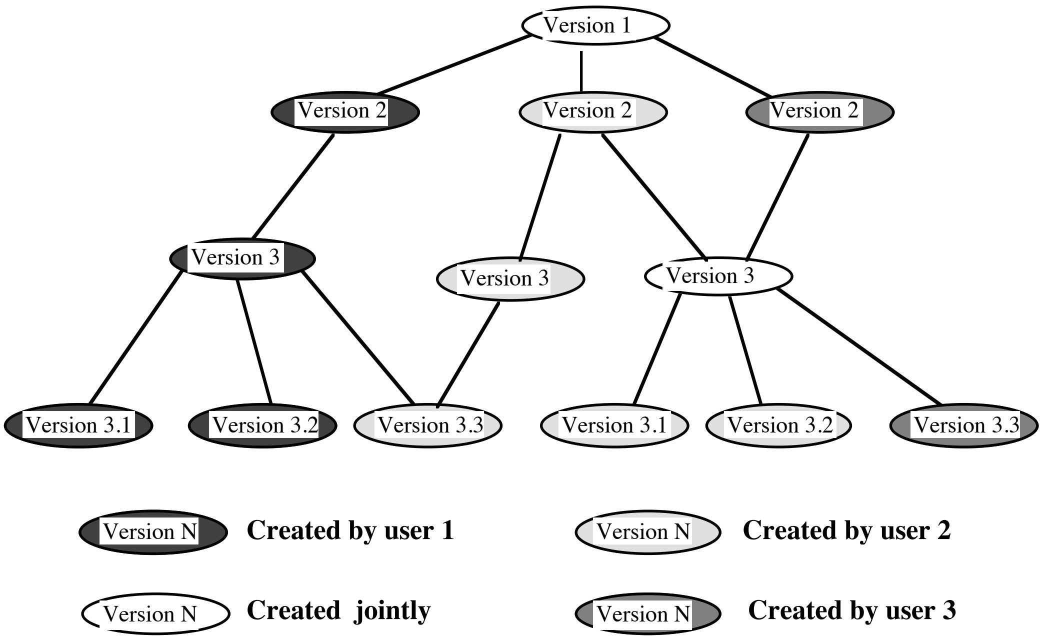


Figure 10.8. The chaos of versions in a cooperative setting.

The solution proposed here is to focus on the most significant items of agreement within thegroup and distinctions across the group in version management. *Versions* are agreed by a group and new versions of an item will act as the current form of the object for normal use. In contrast,*variants* signify a difference in opinion and represent more exploratory alterations in items. Variants do not require any consensus to be created and cannot act as the object.

The role of the object service is to *support* the management of versions. Rather than impose a particular version policy on users the service simply provides a means of recording the version information in order to allow users/ applications management the evolution of different versions. It is important to record:

- What the current version of an object is
- How the version relates to previous versions and variants
- Who the designated version manager for an object is
- What variants exist and what versions these variants were derived from

A central feature of version recording within the service is the addition of a set of system attributes to objects. These attributes allow a number of specific relationships to be associated with objects. These include:-

- The version controller group for an object
- The version of an object
- The variant controller (or group) for an object

When variants are created within the service the objects are linked with variant links which also encode information about the variant. For example, consider the situation where John, Tom and Wolfgang are using an object. John wishes to amend the document and creates a variant to allow him to work on it. The users have decided that Wolfgang will supervise and coordinate the creation of versions. After a period of time a new version is created by Wolfgang. At this point Wolfgang would like to know who has variants of the previous version and to alert them of the alterations to create the new object version.

10.6.1. Version Descriptors and Version and Variant Links

Version descriptors are used hold relevant information on the version of an object. *Version descriptors* are system defined objects with the following attributes

Version_managers: A list of people responsible for managing the version

Previous version : The ID of the previous version of the object

Version Number: The number of the current version

Variant of: This is set to NULL in the case of the object and to the Object ID in the case of variants of the object

Variant Controller: A list of people responsible for the variant.

Version descriptors are particular system objects and the attribute values can be retrieved using the basic facilities for accessing objects. In addition to having version descriptors associated within them versions and variants are linked using *version and variant relation objects*. Each of these have a number of attributes associated with them including:-

Date_of_creation: The date the version or variant was created

Creator : The creator of the version

Additional information may also wish to be added to these version and variant links which can be used by the users and applications enforcing particular version strategies.

10.7. History

In this section, we consider the need to maintain a history of the data as it has evolved through time. If we turn to the "Paperwork" study, we find graphic and physical examples of how a document, as it moves through the desks of an office, grows in any number of ways; by having new sections added (using sheets of paper and a staple), by being annotated in various ways (i.e. a stamp is used to add a grid of prompts on the front cover, which various people then "fill in" as the document passes amongst them), etc. The condition of a document at any given moment in time can reflect how far through the office processing it has come, and an experienced eye can tell (for example) whose desk it needs to go to next.

This kind of object and type evolution can be handled relatively simply with the object data model. The issue of concern here is how easily can the activity of "catching up" with the state of an object be achieved within a computerised service. We are really addressing the problem of support for asynchronous interaction. As we shall see later, the use of awareness is proposed to encourage synchronous awareness within the SOS; the use of a virtual reality environment for a populated information space is similarly supporting synchronous awareness. We need to be able to support asynchronous awareness as well; to support collaboration across time as well as space.

To give a concrete example, consider a shared authoring system where users are working on a shared document. Each user has a specific set of chapters or sections they are working on; they may also have related chapters that they need to know have changed or not. They may also have different levels of need; they may only wish to know if a chapter has altered, but not at the level of detail of words deleted and added.

An author goes on holiday or to a conference or a meeting. On their return, they "log into" the shared authoring system; what they require to see now is a "play back", at the appropriate level of detail, of how the document has evolved in their absence.

This historical or temporal information has applications across the SOS; for example, it can be used to support "fading" awareness (see section 10.11). Underlying a Rooms World metaphor of virtual space, it allows us to trace the activities of a user over time (if this is desirable and allowed) i.e. what rooms have they visited recently and in what order.

While time is the fourth dimension and every item of data, information and knowledge in the world possesses this dimension, it is indeed a pity that the traditional data (base) models (i.e. the hierarchic, network and relational) do not incorporate time as a fundamental part of their models. Nor, indeed, do many of the commercially available OODBMS. There is undoubtedly a growing awareness for the need for temporal database systems in both the database world itself (Tansel 1993) and in many CSCW applications and systems development efforts (Rodden 1992a, Kaplan 1991).

Efforts have been made to extend existing data models to incorporate time as a integral part of the model and query languages (such as SQL variants) have been given additional time clauses (such as "when", the temporal equivalent of the spatial "where").

10.7.1. Discussion and Implications

Any temporal store needs to maintain, on a per-object (or whatever the chosen granularity is) basis, four "time variables". These are the start and finish points for "valid time" (sometimes referred to as "real time") and "transaction time" (sometimes referred to as "system time"). If a "fact" about the status of a shared document is added to the system, it's start valid time would be the date of the initiation of the document, and it's finish valid time would be the date of it's completion. It's start transaction time would be the time when the fact was added to the system and it's finish transaction time would be the time when the fact was "deleted".

Temporal information can be "grafted on" or layered above an existing object store. For example, an type "time" containing the four time variables as attributes could be added to the system. Then, objects could be linked to their temporal information. This means that queries involving time have to be expressed in such a way that the “time clause” can be directed at the linked time objects. It may be possible to use adapters in such a fashion. It is possible to visualise a “history adapter” whose function is to store the temporally altered value sets of attributes of a “base” object whose state always reflects the “current” snapshot state of the object. By accessing the history adapter it should be possible to recreate the state of the object at any given point in time.

In order to make some progress on the SOS specification, the following assumptions are made :

- the granularity of a time specification is on a per-attribute basis
- every object requiring that history information about itself is maintained has a history adapter associated with it. A history adapter contains the four attributes “valid_start”, “valid_finish”, “transaction_start”, “transaction_finish” for each attribute. Moreover, multiple values are stored for each temporal attribute. The main object itself always represents the “current” state of the object.
- the Query Specification object is capable of including appropriate temporal clauses delimiting the range of a query to a set time period. Using the combination of the main object and the history adapter, it is possible to generate a “temporary” object which represents the state of the object at any given time (or during any given time period).

The “Query Engine” of the SOS needs to be able to cope with temporal queries. The generation of an object history requires the identification and recording of significant actions. This can be done within the shared object service using a special form of event transformer which transforms actions on the object into *history events* which are added to the recorded history for the object. Consider an object which represents a file shared between Tom and John. In terms of the object the only interaction either user has is by invocation of methods such as read and write. However, by specifying a sequence of these actions we feel meaningful we can log higher order history events. For example, we could decide that a sequence of the form write(tom), close(tom), open(john), write(john) was a significant transfer of editing to john and record it as tom passed object to john.

Event transformers are general purpose event handlers which from a set of rules can translate a sequence of low level events into a higher order event. History events and the use of event transformer are a specific instance of this form of service. A more detailed discussion of events and event transformers is given in section 10.

10.8. Access

Multi user systems have used access mechanisms to control the different action of users on the shared systems. A variety of different models exist which offer different facilities to different application communities. Most of these access models are derived from the Access Matrix Model. In this general model there exists a set of passive Objects which are the resources to be protected, and a set of active Subjects who wish to access these objects. Each Subject, Object pair (S, O) is associated with an access right(s). An access right is a privilege that a subject (S) has over an object (O). A Subject may access an object in a way specified by the access right(s) held by the pair (S, O). A number of more specific models are derived from the Access Matrix including:-

- The Access control list, which divides the matrix into columns, each user holds a list of accessible objects. The addition of a new object requires each users' access list to be updated
- Capabilities: here the matrix is divided into rows, and each object holds a list of users that may interact with it. It is difficult for a user to tell which objects they may use. Addition of a new user requires all objects to update their lists, and the list of users able to access an object may become large.

Although the access matrix defines the principal model for the control over a users (or subjects) access to an object, it faces a number of problems when used in a multi-user collaborative environment. Problems with the above model which have been highlighted include -

- It does not provide faculties for the management of roles or groups, individual users tend to be used and dynamic amendment is difficult to do.
- It does not facilitate easy specification of access rights
- It does not offer a mechanism for specification of the access matrix, or for access checking

In addition, most access models adopt a static view of access control. The assumption is that access is sent up and only occasionally altered by a single administrator. However access models within CSCW system must also allow a number of additional facilities

- They need to be more dynamic with access altering frequently
- Access alteration been inferred from the state of some activity or a users role.
- Facilities to allow a group of users to manage changes to access.
- The need to provide mechanisms finer granularity of access control than that offered by the access matrix.

The approach adopted by the COMIC shared object service is to use a specialisation of interface adapters which provide access management facilities. Access adapters contain a table of interface users and set of rules which determine the access of users to the object. The access rules within the adapters also make use of specialised state information to determine access. This state information can be used to represent the external activities surrounding the shared object store.

Traditional access models also exploit a set of relationships between users and the shared object they are accessing. However this relationship has tended to be limited to considering only object owners, members of a specified groups and all others. The access model within the shared object service allows more sophisticated semantic attributes to be defined and used. In particular, the access model allows each object to have the following attributes which can be used within access rules.

Owner: The person or group who now owns the object

Responsible: The person or group who is currently responsible for the object

Possessors: The person or group who are in current possession of the object

Moderator: The person of group who moderate access to the object.

Creator: The user who initially created the object

Given the vast array of potential relationships with the object and potential access to operations recording access information is a problem. Particularly given the need for dynamic alteration of access. To overcome these problems we make use of adaptors to codify access rights. Adaptors contain an *attribute comb* which determines the visibility of attributes through the adaptor. Attributes when viewed through the adaptorare in one of three conditions[1].

- Invisible
- Visible but not editable
- Visible and editable.

The adaptor holds a table of these presentation conditions for attributes. Each attribute has a different condition entry within the table and the mechanisms to set and retrieve these conditions are provided. In addition, access rights to operations on the underlying object are classified within the adaptor into particular access levels. This classification allows users to be granted a particular level of access and to move between levels as necessary in a dynamic manner. The adaptors manages the mapping between access level and the invocation and with reference to an acess manager verifies access to the underlying object.

10.9. Queries and Views

While we do not believe that the SOS will be a fully functional database management system (however desirable some of us might find this), it will need to supply some of the functionality of such general systems. Already in this document we have discussed the SOS “Query Engine”; this will be a server of the SOS capable to interpreting query expressions and returning a list of object identifiers which match the expression for subsequent processing.

Clearly, there will be a number of mechanisms available to the end-user of the SOS to allow them to search, locate and find out about objects within the system. The VR component of strand four for example postulates populated information terrains where users can move through a virtual realisation of the data space. To some extent, this can be interpreted as three dimensional browsing. There may be two dimensional browsing, such as an organisational knowledge browser, such as those suggested in strand one. Even in the VR realisation, however, there is discussion of user agents : intelligent objects which rush round cyberspace looking for objects that provide certain services. Such an agent would surely rely on underlying

1 Actually, we are not sure there is any sort of limit. This is only an initial set. We could imagine having a set of different modes which require different permissions to edit or specified different patterns of propagation.

SOS query mechanisms. So there is a need for a query mechanism both as an end-user interface and for agents within the system to directly locate sets of objects.

The use of the term “view” in this section refers to (relational) database technology rather than any presentation issues. A view in relational database systems can be specified as an SQL query which operates on current relations (or views) to produce a new view of the data which can (within certain theoretical and practical limitations) be manipulated as if it were an ordinary relation. The SQL query breaks down into a sequence of relational algebraic operations (typically select, join and project). While object data models do not currently have a standard, much research in providing ODM algebras examine the possibilities of including relational style operators such as the ones listed above.

It is clear that one of the problems of the relational model is that it splits conceptually singular objects across a sequence of relations; in many cases, the algebraic operators are used to “reconstruct” the singular object. If the designer of the object types has done their job properly, there is no need for this “reconstruction”. However, we cannot ignore the need for unforeseen “ad-hoc” queries, and it is this other use of the operators that ODMs require. Therefore, we also believe there is a need for views within the SOS.

Within the SOS, a query specification will be modelled by an object. Such an object will consist of a list of object “facets” against which will be a set of required values and an arithmetic relational operator (i.e. equal, not equal, less than, etc.). Facets will also be related by Boolean operators (and, or, not). Such a query specification (with the matching “query engine” provided within the SOS) should give us the computational equivalent of general ad-hoc query mechanisms within standard database systems.

Within the SOS the result of a query is itself an object. If it is a collection, then its state is represented by a set of Object IDs. If we have issued a query against a collection of objects representing information about people, then if the query was “get me all people whose first name is Ludwin”, we will have a collection of IDs whose objects have the attribute “first name” set to “Ludwin”. We can keep that collection ID around for a while, long after the query was first issued. If this SOS supports persistency (which it should), then we can return to the “Ludwin” collection at some time in the future. In this case, what happens to the “Ludwin” collection if a new object with name = “Ludwin” is added to the base collection that the “select” operation was carried out on in the first place?

There may be a need to specify views as transitory (the resultant collection is discarded as soon as the view operation that brought it into being is out of scope of the surrounding transaction) or persistent (the resultant collection is saved and updated if new objects are added which match the view condition). To support the dynamic addition (and indeed, deletion if some object changes the value of their first name attribute) of objects to a collection, we need some sense of a “future” query, and this, we believe, can be handled by the event mechanism described elsewhere in this document.

10.9.1. Using Adapters to support DBMS style operations

In this section, we briefly examine the possibilities of using the object adapter mechanism proposed in this document for implementing DBMS style operators. We have already argued that while the SOS does not require the full-blown functionality of a DBMS, many DBMS style features (such as retrieval by query) are both desirable and necessary. This implies (for example) the need for a "query engine" for the service capable of handling such requests.

Turning to the need for relational-style operators (typically to support view specifications), we believe that these can be modelled on the use of object adapters.

A number of alternative mapping are possible:-

- An interface adapter can rename object methods depending on context
- An interface adapter can reduce the number of methods presented by an object
- An interface adapter can aggregate a number of objects to provide a single interface.

Thus the general structure of an interface is as a table of mappings between method access points provided by an objects interface and the presentation of these access points to users. It is envisaged that the use of interface adapters will override access directly through interfaces. The different alternative arrangements for interface adapters are shown below in figure 10.9

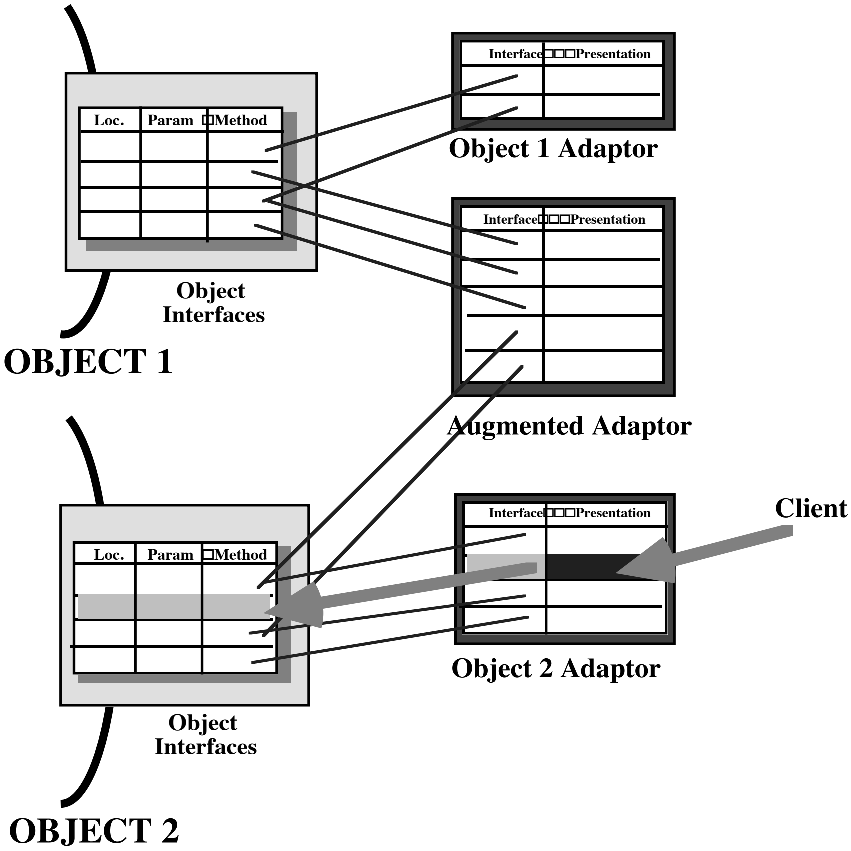


Figure 10.9: Mapping Interface adapters to Object Interfaces

In this figure, objects 1 and 2 are examples of base objects. The object 1 and 2 adapters can be thought of as the result of a projection, limiting access to attributes/methods in the base objects. The augmented adapter which allows access to some attributes/methods in both objects 1 and 2 can be thought of as the results of a join.

10.10. Events and Subscription

The event mechanisms within the COMIC shared object service enable users of the shared object service to be aware of the actions of other users of the services. This section considers the parts of the shared object service which provide for the generation and handling of events.

The feature which most distinguishes the COMIC shared object service from other multi-user storage systems is that the COMIC service has as an explicit goal the need to provide an awareness of the action of others on objects within the shared object service. Thus, in the COMIC shared object service when an object is accessed or altered the systems should inform the relevant users the action has occurred in order to promote awareness across the service.

In addition, the shared object service should provide facilities to allow users of the shared object service to make others aware of actions of interest to them. The mechanism used to provide this form of awareness within the service is events. The shared object service provide facilities to allow events to be defined, created, related to other users and handled in appropriate ways.

Within the COMIC shared object system all events are of a specialised class of system objects These object as in the case of all other objects within the service are shared across the community of co-operating users. The distinction is that a set of specific services are provided to handle event objects. These services inform concerned users when a specified action occus related to event objects

Two different kinds of event are significant to the shared object service:-

- The need to make public the alteration of objects
- The need to allow actions outwith the shared object service to make themselves public.

As an example consider the situation where a number of users are working on a shared drawing. One of the users needs to wait for another user to complete a specific tasks within the shared drawing before she can continue. In this situation the user wishes to focus on another task until her colleague is finished, however she wishes to be informed whenever the drawing is amended by her colleague. Within the shared drawing service the user wishes to be informed whenever the object associated with the shared drawing is modified.

In contrast, consider the situation where a development engineer needs to wait until the drawing is complete before proceeding with a particular set of actions. It is not possible to determine the completion of the drawing solely from the state of the drawing object. Rather, the users working on the drawing must make public the knowledge that the drawing is completed. In this case users would create an event object which would stipulate that the drawing was complete. The creation of this event object would then signal the waiting engineer that the work can then begin on construction. Note that the mechanism here is primarily descriptive allowing the users to make public knowledge about their action rather than a prescriptive one which would as part of the mechanisms prohibit the development engineer from proceeding.

In supporting co-operation among a wide set of different users and platforms, possibly with different interfaces to the applications and diverse language bindings, a uniform and clean event mechanism is needed. In the SOS and *event handler* provides a central facility responsible for the handling and support of requests and subscriptions on events.

An event is generated by a number of different actions within the shared object service. Events can be generated through some change in an object in the SOS server. The event could alternatively be constructed through an event generated by an database entity directly or by an event-adapter that encapsulates the database entity. The passing of the events between different nodes is accomplished by (logically) sending the events to the Event Handler which distributes them to interested participants. To fulfil its responsibilities the event handler builds upon the event services supported by the interface and interaction system used by the platform.

The event object describes an action,. Because the actions can be of different kinds, SOS needs to support different types of events. The set of all types of events will form an hierarchy where event subtypes are modifications of their parents, and they will share their parents' properties with all of its subtypes. This means that event types from the same part of the hierarchy will share some properties and be comparable with respect to those properties.

The shared object service relies on facilities provided by a *Subscription Handler*. The subscription handler is central to the services that support users' (and objects') awareness of each others' actions. This portion of the service maintains a record for the clients of with services and events or types of events are of interest to them.

10.10.1. Event Subscription

A client, application or user, could declare interest in a certain event category by declaring an interest at the subscription handler. The subscription mechanism spans from sharing to awareness. With this we mean that one can be informed about changes of other entities and also subscribe to other users' awareness events at a higher level of complexity.

10.10.2. Basic subscription functionalities

Lazy vs. eager subscription. The client objects can subscribe to an event in an eager or a lazy fashion. If an eager subscription is used then the client is informed about the event as soon as possible. In the lazy awareness style the client is just updated when it tries to access the changed entities. An example is a document where a change is not of any interest to another user until she accesses a related document. In a tightly interactive situation between different users perhaps a much wider set of events is of interest..

Anonymous vs. public subscription. One can declare an anonymous subscription to certain types of events, i.e. without making that information available. The other, public, extreme is that everybody can identify the user as a subscriber. There should also be possibilities to make subscriptions public to certain groups of users and hidden to others.

Subscribing on specific categories of events. To make it easier to subscribe on a wide set of events the clients can also declare interests in a whole class of events and in events that have certain properties.

Announcing events (broadcasting, reverse subscribing or public relations). In some way he different actions that could be of common interests must be announced. A change is normally announced through the triggering of an event at the Event Handler. If the event one wants to send is of the type, that there will be no more changes to an object then one has to use the mechanism of announcement. These events represent meta-information about the object.

Declaring and retracting subscriptions: An object announces an interest in other objects or events by declaring an interest of the object at the subscriptions' department or directly to the objects of interest, as a indirect access to the subscription department. A retraction to the objects or to the subscriptions department is also supported.

Subscriptions on system events. Typically many events occur as external events, i.e. outside the scope of the SOS software. To make those events fit into the rest of the SOS-environment as transparently as possible one can subscribe to systems events through the subscribing department, which is responsible to transform the event into a suitable form.

Security and prohibition of subscriptions. The mechanisms that handle subscriptions must besides being robust, handle situations of security. For instance one does not want unauthorised people to read secret information, or even be aware of that such information is distributed. This means that the subscription department must allow users to prohibit objects from respectively allow others to (perhaps a subset or group) know that certain actions take place. The object oriented style of SOS should make it easy to change and add new functionality, e.g. for security, later.

Subscribe to other subscriptions: can allow subscription of other subscriptions, e.g. a system scrutiniser that subscribes to the adding or removing of subscribers to a specific object. A user declares interest in knowing about other subscriptions in the same way as one normally makes other subscriptions. This mechanism can be essential as support for different kinds of awareness, e.g. if one wants to be informed about other users' activities and interests (with due considerations of integrity, a social rather than technical issue).

10.10.3. Event adapters

The system will provide a number of mechanisms to achieve awareness of other participants actions when they are using SOS. The information about others' use of SOS is collected into the event handler, and further handled by the subscription department. The entities the event handler are dealing with are, as the name indicate, events.

A mechanism for describing generation of events is using so called Event Adapters. This is a type of Interface Adapter that does not change the behaviour of an object in any way. In contrast to for example Locking Interface adapters, the only behaviour it has is to generate events when the object it adapts is used, as described in fig xxx. This means that when one sends a message to an object within SOS it will first reach the Event Adapter, that generate an appropriate event, and then the message will be passed to the adopted object, or the next Interface Adapter.

The use of Event Adapters means the one can on per object bases decide what event mechanism is needed, and further more, because the Event Adapters are themselves first class objects within SOS it is possible for the application programmer to design his own Event Adapters when needed.

An object adapter could in turn be encapsulated in an event adapter that informs others of changes in the object. Event adapters are a means of realising the event services within the shared object service. An event adapter knows about subscription relationships. If an object declares interest in events to or from another object or agent then an event adapter "filters" events to and from the interested object (see figure 10.11 below). If an event is of "public" interest that event is propagated further.

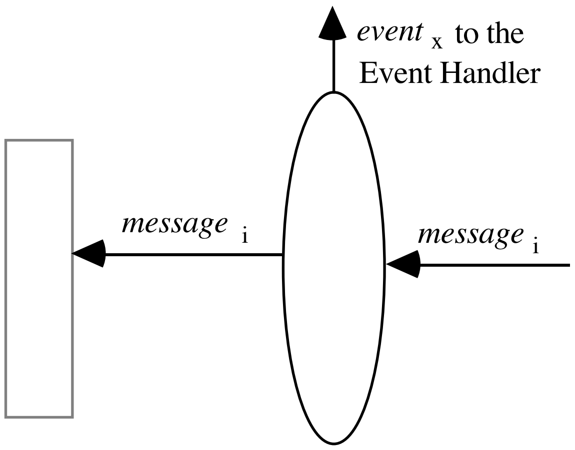


Figure 10.11: Event Generation using an event adapter

Event hierarchy

Events within the shared object service are objects in their own right. Each event is an instance of an object type which forms part of an inheritance hierarchy. The hierarchy uses abstract and concrete classes of events. The concrete classes are used by an object to describe a message received or an announcing event. The abstract event classes are not directly used but they describe similarities between objects of there subclasses. This means that for instance an abstract event from an input device can act as a abstract input device and by that be used instead of the concrete subclasses if the functionality of the abstract device serves the purpose. This means that the event hierarchy will try to lift as much information as possible from the concrete classes to the abstract classes to make it easier to communicate when theevent classes used do not match completely.

Event Transformation

In some situations it is not enough to use abstract event types as a way to communicate between systems that uses different event types. In those cases one need to transform one event or a sequence of events to another event or sequence of events, to be able to understand each other.

Certain events can be transformed into other events that better fit the application's requirements at a certain node. The demand for transformation of a certain event type is registered at the event handler, which transform given events into higher level events from a set of given rules. A set of given event transformer objects can be tailored to meet users' and applications' needs by:

- extending the hierarchy of event transformers.
- change parameters of existing event transformers
- clustering a set of existing events and define the cluster as a new event type.

Event transformers are also the principle means used to generate object history records. A set of given object events are aggregated by event transformers to define history events which are added to an object interaction history. A further significant use of event transformers is to support the display and update of shared objects across remote screens. Event transformers collect atomic user interface objects to form shared object service objects which are then propagated to other user displays. For example consider the object been share between two displays in figure 10.12. User interface events in interface A are transformed to shared object service events which are then send to the transformer associated with interface B and reported as appropriate interface events.

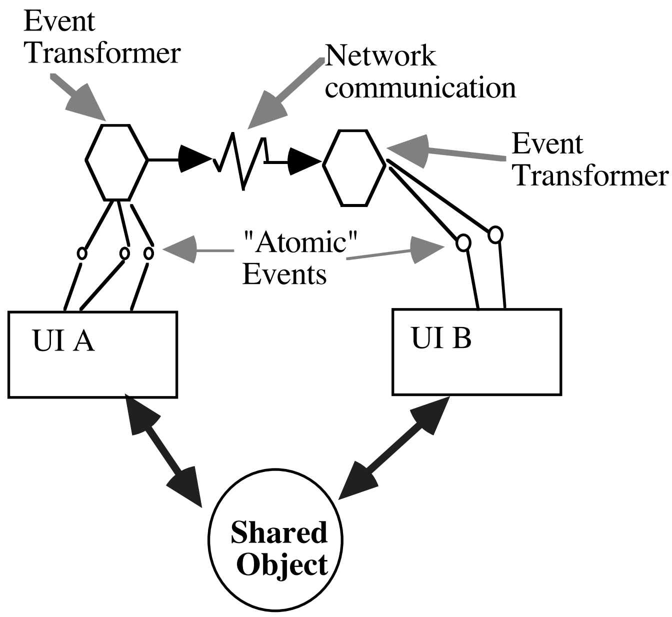


Figure 10.12 : transforming between interface and shared object events

Events are one of the principle mechanisms used to provide an awareness of co-operation within the shared object service. Events can be used to categorise and generate an indication of cooperative usage and to propagate this to other cooperating users. A significant aspect of the shared object service is in the promotion of awareness generated from a consideration of events within the service.

10.11. Awareness

One of the key concepts identified early in the COMIC research has been the issue of awareness. Described simply, users need to be aware of each others presence and activity within the system in order to initiate collaboration. There have been (initially) two approaches towards this topic; one, the object approach (Mariani and Prinz 1993) considers objects within an information store (typically a database system) and users browsing or generally accessing the objects within the store. The spatial approach (Benford 1993, Fahlen 1992, Bowers 1993) considers a VR setting.

The object approach uses a variety of formulae to calculate an awareness factor between two users; this takes into account both "space" (how near are two objects to each other : here, again, various formulae and object classifications are used to provide this spatial metric) and "time" (how long ago was a user accessing a certain object : this is used to provide the fading of awareness over time). The spatial model directly represents users as items within VR and these items possess the attributes "focus" and "nimbus" (so, too, do objects which appear within VR).

This focus on awareness allows an integration of the object and spatial approaches to managing interaction and this theme is continued in the final chapter of the deliverable. For the moment, this section will concentrate on how awareness is supported across the object service. As mentioned at the start of this document, object awareness is considered to be in the province of service agents associated with core shared object service in the form of awareness handlers. However, the information upon which the awareness factors are calculated is held in the SOS by the shared object server event objects. A more detailed consideration of the calculation of awareness within the shared object service is given in the following section.

The key mechanisms in promoting in the use of events. Events objects are potentially generated by every "system call" made to the SOS. Associated with every event is the following information : the ID of the user who issued the system call, the date and time at which the call was made, the IDs of the objects involved in the execution of the system call etc.

Based on the information held in the event objects, the service agent (the Attentiveness Manager) can calculate attentiveness factors between objects and thence between users (of those objects). Until now, it would seem that there is little that needs to be added other than concern over the attributes of the event objects. However, it must be taken into account that not all objects are created equally.

Some objects may possess a greater "pay attention to me" attribute than others. In the VR example it is simple to think of an example; a "megaphone" object, when used, should be heard by everyone within its nimbus. In an object approach, used to support a groupdesk style application, perhaps a memo object from the boss should have a greater "PATM" attribute than a scrawled note from a colleague (or perhaps not!).

We therefore suggest that every object possesses a PATM attribute and accordingly methods to get and set the value of that attribute. The impact of supporting awareness (in the manner described above) is the following :

- for every SOS "system call", an event object must be generated.

- to support “fading attentiveness” and, perhaps more importantly, asynchronous attentiveness, event objects must “persist” over time (theoretically, be “immutable objects”). This “immutability” should also apply to the objects referenced by the event objects (otherwise, interpretation of the event objects by the awareness manager becomes nonsensical).
- by placing the responsibility of calculating the attentiveness factors between users in service agents, we hope the overhead of said calculation does not impact on the SOS itself.

An associated issue in promoting awareness is the use of history to allow an asynchronous awareness of object use. History within the SOS is generated as a result of spacialised event transformations. The close coupling of events with history generation allows use to specify an awareness of history which can also be calculated by awareness handlers. This awareness of history allows users to be aware of the action of users in a time independant manner.

10.12. Interaction and presentation

SOS, the shared object service, provides support for the creation and management of shared objects which are intended to be used as an integral component of cooperative applications. A crucial factor in these application is the mechanisms used to support interaction with this shared information. The majority of these mechanisms are provided as part of the shared interface service associated with the COMIC shared object service and described in a forthcoming COMIC report

It is essential that the shared object server utilises the shared interface services in an appropriate manner. This requires the shared object service to provide a set of links to the shared interface service. In addition, the shared interface service may need to allow direct access to shared objects and requires SOS to incorporate facilities which support object manipulation. Often this manipulation will take place in a time independent manner and the need for the synchronous services provided by the shared object service is less obvious.

This section briefly reviews a number of important interaction and presentation issues that have to be handled by the shared interface and/or the shared object service.

10.12.1. Different presentations of objects in SOS

Objects in SOS can be handled differently by different users and applications given the nature of the work they support. For example, experts in different fields may wish to look at an object in different perspectives suited for their domain. When various views on an object are used there is also often a need to have alternative ways to manipulate the object and the invoke the operations of interest to the particular users. The use of different object presentation and manipulators on objects in SOS is entirely handled by the application or some interface service. To allow this to occur the SOS needs to support the development of appropriate links between the object presentation and the underlying object within the shared object service.

10.12.2. Supporting awareness of similar objects

When people access and use a similar set of objects within the service they need to be made aware of this fact. This requires the shared object service to present this awareness to user in an appropriate manner. To do so requires the shared object service to publicise the awareness of other users. This can be achieved by the use of a set of appropriate events and a calculated awareness metric (cf sections 10.10 and 10.11).

To facilitate cooperation from this awareness the shared object service should link to mechanisms to support the direct communication between users. If users of the service are working together on a shared object this communication is vital particularly in the case where database objects provide different visualisations. Many of these facilities are provided by either the underlying distributed platform in the form of multimedia communication facilities or the shared interface service.

10.12.3. Supporting awareness between participants

In a situation where people work together in a close way they need to be aware of the actions of each other at least if it influences them. In some situations there is enough to see the changes made, when they are accessed the next time. Other situations require changes to be viewed and highlighted. It is often useful to have an overview of what other people are working with, regardless of the representation on the screen. An example is someone studying a users recent work which is still incomplete. The user will often want to be made aware of this to offer the possibility of directly contact the observer to offer an explanation.

A significant feature of the real time sharing of objects is the need to support the reference to objects across a group of users . One solution to the problem is to use some form of telepointer or a highlight operation, so that the an object of interest is highlighted across a range of interfaces and applications regardless of view used. In order to support this kind of information to be passed between applications one cannot directly use the SOS events but must rely on the set of events provided by the shared interface service.

10.12.4. Supporting browsing

In many cooperation applications it is of crucial value for participants to be able to browse among objects in the application space, with or withourt representations of other users. An example is given in the social workers example in section 9.4.4. in chapter nine, "If all the world's a stage…", where it is pointed out that browsing facilities might be especially useful in "less structured" areas of that kind.

To support such browsing the shared object service needs to uphold some, at least partial, metric or "nearness" between objects and some history, or at least marking of cahnges, since the last time an object was visited.

10.13. External References

This section addresses a rather subtle (but important) point: not every object of interest can be stored within the SOS. Particularly when we consider (say) a journal, book, or set of conference proceedings. The fundamental issue here is that we believe it is important that these “real physical” objects can be subject to similar kinds of operations as the objects stored within the SOS. For example, just as we can issue queries regarding stored objects, it would be nice if we could issue queries regarding real physical objects i.e. “Who currently has a copy of the proceedings of ECSCW´91?”.

From even a cursory examination of the studies described in chapters one and two the hetrogenious nature of documents is apparent. There is no reason to conjecture that this hetrogeniety will diminish in the case of electronically based objects. In contrast it is likely to increase with a continued need to access paper objects outwith the shared object service. It is also important to realise that objects stored within the SOS may make reference to “real physical” objects every bit as much as they may make references to other objects stored within the SOS. For example, an electronic catalogue of review forms for a journal may make reference to both electronic and paper instances of the forms sent to the journal editor. Such “real physical” objects can be considered as external (outwith the SOS) objects.

In some senses, an object which contains a reference to an external object can be considered as being incomplete. Unlike a fully internal object, we do not possess all the information “at hand” to display, manipulate, indeed generally consider the object. This impacts on issues of how do we display or present an incomplete object, how to we query and retrieve an incomplete object, etc.

The closest analogy we can find (albeit not a very good one) is to consider the functionality provided byt archival systems. In we considered fully internal style of SOS with limited storage space the need to provide some archiving of information becomes crucial. As objects get older (i.e. the time since the object was last “used”), they can be archived into tertiary storage (i.e. mag tape). However, the references to the now archived object remains throughout the SOS. When a user accesses an object which contains a reference to an archived object, the system knows that it has been archived and directs the operator to mount the appropriate tape and will then retrieve the object from tertiary back into secondary storage.

This is not a good analogy as even the archived object, while (in a sense) external to the current state of the SOS, remains electronically retrievable and can be reinstalled. An incomplete SOS object can readily (albeit with the passage of some time) be restored to completeness. Many of the external references discussed in this section are considered to never have been a part of the internal SOS. Broadly speaking a number of classes of external object exist.

- Electronic objects which are currently not required and have been removed from the SOS and archived.
- Electronic objects which are detached from the SOS and are not readily accessible. These objects are characterised be been currently relevant but diffi-

cult to access. For example, the object may have been placed on a portable or mobile computer.

- Non-electronic objects which are related to the SOS and have some form of electronic manifestation within the service, However, the object itself is not part of the SOS.

How do we choose to represent these external references? One way might be to actually have a “pseudo object” which is actually stored within the SOS but which simply contains details about the “real physical” object. In a sense, though, this is no different to how we might represent people within the SOS. Another approach might be to have a simple text field which contains the name of the real object (rather than the ID of the pseudo object). It might be nice to “tag” such fields with the “type” external reference.

If we choose the latter approach, we face tremendous naming problems. People may chose different labels for real objects i.e. “the ECSCW´93 Proceedings”, “Proceedings of ECSCW´93”, “ECSCW´93, Milano” etc. The forms may be even more widely differing that this. This impacts on how we might query the SOS for objects containing particular external references. (Of course, this holds true for any textual label that the pseudo object might contain so it is not really just a problem of the label approach)

One answer to some of these problems is to admit that incomplete objects will always be incomplete! The purpose of the external reference is merely to give us a hook into the physical objects, where presently none exists. If we are searching for a copy of the ECSCW´93 proceedings if we can query the (incomplete) objects which reference the proceedings we can hope that we may find some clues as to whose office currently contains them. Event and history considerations can also assist us; we may be able to find out who recently has been searching for the proceedings (just as we are) and perhaps that person was successful in their search and they might be a good starting place.

We exploit orthogonality across the service as a means of aiding our representation of external objects. We have chosen to adopt a specialised form of adapter as the means to represent external objects (Figure 10.13). External adaptors act as the electronic manifestation of the externally held artifact. The role of external adaptors is to provide a bridge between objects within the SOS and those within the real world which are relevant to the SOS.

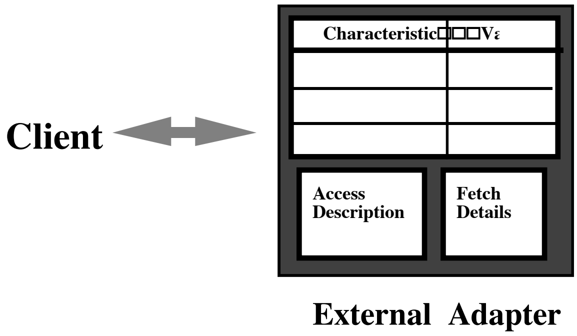


Figure 10.13: An external Adapter

On invocation rather than directly return a result the external adapter informs the client that the object is external and offer the possibility of access information to the external object. This access information have has number of forms:-

- Details of how to retrieve an archived electronic object.
- Details of how to establish electronic commnication with a detached object. .
- Descriptions of non-electronic objects as a set of textual labels and a description of how to access the external object. This may in turn involve the description of some particular organisational policy or procedure. The role of this access description is similar to the use of procedure descriptions within the organisational browser described in deliverable1. The description acts as a user resource for accessing the external information. .

As in the case of access rights external accessibility information is held as an integral part of the external adapter. Two types of access information is maintained. The access description for use and inter By using external adaptors which can act as psuedo representation of objects within the SOS queries across the database can be handled in a orthogonal manner. In addition, queries can also access external objects and query textual labels whithin the adaptor

10.14. Conclusions.

This chapter has described an initial set of services which need to be provided by a COMIC Shared Object Service. The intention is that this document will be used as the basis for a more detailed service description including definitions of interfaces for each service and their functionality. The provision of these facilities will have a number of architectural implications and will need to be sensitive to issues of performance and distribution. The facilities presented here have met a number of specific criteria.

- The identified facilities augment rather than replicate the support for objects provided by distributed infrastructures such as ODP and OMG.
- The service maintains a clear seperation between the behavioural semantics of the objects and the management of the sharing resulting from their context of use.

- The facilities have been informed from a series of empirical studies of the use of paperwork within organisations.
- The facilities provided are semantically lightweight in that they make no assumptions about object use. Rather they intent to provide an appropriate set of mechanisms to allow users to cooperatively manage the use of shared information.

The following sections of the deliverable build upon the facilities provided within the shared object service by examining how they can be used to promote a sense of wareness across a community of users and to support a particular cooperative application. Finally, the use of the shared object service facilities to support the management of cooperative activities is discussed before considering the relationship between the shared object service and the spatial model presented earlier.

11. Awareness in Collaborative Object Systems

Wolfgang Prinz
GMD

J. A. Mariani
University of Lancaster

This chapter describes an approach to provide awareness about users in a cooperatively used object system. Such an object system could be supported by the shared object service. Some possible implementation solutions are proposed.

11.1. Introduction & Motivation

There can be little doubt that object systems, which continue to gain popularity and wide-spread usage, are the wave of the future. Object storage servers, such as the current generation of Object-Oriented Database Management Systems, have been developed to cope with the new demands on such systems by application areas such as software engineering environments, CAD/CAM systems, office information systems etc. To some extent, these new application areas can be classified as examples of CSCW (Computer Supported Collaborative Work).

In order to help and encourage collaboration, systems need to provide increased user awareness. We shall see later in this paper a number of scenarios where users who are sharing information would usefully be aware of other users accessing the same, or the same kind of, or interelated (in some way) data. Once users are made aware of each others presence in the information space, there must be simple-to-use communication mechanisms available to support collaboration.

In this chapter, the emphasis is on synchronous awareness. While more than one user is accessing the information space, they can be aware (to some degree) of each others movements and activities, and communication between them can be initiated. However, it is important to bear in mind that asynchronous awareness and collaboration is just as (if not more) important than synchronous awareness. This requires some kind of history mechanism to be supported by the object storage server.

Lastly, privacy plays an important role within any attempts to increase awareness. We merely note it at this point, and somewhat over-confidently predict technological solutions to this problem i.e. perhaps you can only be aware of someone (or something) that wants you to be aware of it. Nevertheless, this is an issue that should (and will) be investigated through the use of the prototype system.

In the chapter, we will consider a "generic" Collaborative Object Server (COS) and propose solutions to the various levels of user awareness in terms of this COS. In one sense, as each solution is proposed, we are in fact building a set of requirements for any specific COS which would like to provide these levels of awareness.

The chapter describes several steps for the provision of awareness. Each step is described in one of the following sections. Each section describes general considerations first and then an implementation solution for the COS is given. We begin by considering the widest possible granularity, that of providing a Unix-like "who" command.

11.2. Awareness at the system-wide level

11.2.1. General Considerations

The system should inform a user about other users who are currently using the system or who has been using the system. A straightforward implementation of a Unix like "who" command is of course not reasonable in a wide area distributed system with hundreds of users. Therefore, filters are needed which allow a filtering of only those users which are of interest to a user. This filter can be build using the following criteria which can be provided:

- manually by a user defined list of those users that the user is interested in
- automatically and dynamically by using a query expression. For example, in an organisational context, selecting all users who belong to the same domain, i.e. project, department. These can be partners of international projects.

The presentation of the currently active users should provide means for contacting any of the other users by various communication means. A selection of appropriate communication media can be made dependent on the available communication facilities. A spectrum from simple Unix talk, or e-mailing a message to audio or video-conferencing is possible.

11.2.2. Possible Implementation

We can expect that whenever a user starts to use the COS through one of a number of client interfaces that their presence within the system is registered. The period of time during which a user is "logged onto" the COS is modelled as a session. A currently active user could be modelled using the following object type:

```
objectType  Cos_User
{
Attributes:

        userName                    the distinguished name of the user
        userHost                    the host where the user is located
        active              true/false depending whether the
                    user is still active or not.
        sessionStart        timestamp when session was opened
        sessionClose        timestamp when session was closed
        closeState                  indicates if the session was closed correctly
                            or if an exception or error occurred

Methods:

        isActive()
        getUserName()               returns the user name
        getUserObject()             returns the user object
        getUserHost()
        getSessionLength()
        getCloseState()
}
```

For each session an object of this type is created and included in a list called “UserList” that stores all of these objects. When the user closes the session (or “logs out” of the COS), or when an unexpected exception occurs that will finish the session, the status of the object is changed from active to passive and a timestamp is stored in the object. This allows us to keep a usage history of the COS.

In order to present this new information to the user an appropriate interface must be developed. After a session has been created, the UserList is scanned for all active users. These are shown in a new window that lists the names (or icons or photographs, depending on which client interface is being used) of all active users. In addition, the window allows the selection of a user in order to contact him by selecting a communication medium.

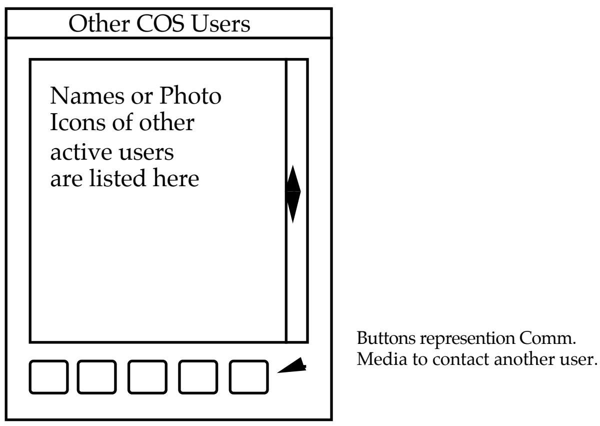


Figure 11.1 : Possible Visualisation of a “who” window

The list contains the name of a user, the host where he is currently located and his login time.

The following communication media may be offered:

- unix talk
- e-mail to a user
- audio
- video

We can display the history information i.e. a list of all users who used the COS in the past. The information per user is then extended by the logout time and by information on how the session was closed (normal or by error). It must be checked how this conflicts with privacy issues.

It is obvious that the user list can be easily retrieved when new session is created. However, how is the display updated while the session is running? A straightforward solution can be achieved by periodically reading the UserList from the database to update the window display. This use of polling is a very specific solution to this type of problem, and we believe a more general mechanism is required; one that can be applied to a wider range of situations. This can be done in form of an event notification service in the following way.

An event object type is defined that implements the basic functionality for event objects.

```
objectType  COS_Event
{
Attributes:

        eventType                  // type of event
        user                       // user who caused the event
        creationTime         // timestamp when event was created

Methods:

        getEventType()
        isEventType()              // return true if eventType matches
        getUser()
        getCreationTime()
        moreRecent(time)     // returns true if this event is
                                   // younger than time
}
```

This type should be inherited for further refinements, for example when an object needs to be included into the event object.

11.3. Awareness at the object level

11.3.1. Introduction and Issues

Assuming that object services will become widespread and that it will be normal to be logged in, then the simple “who” information described above will not be sufficient. A finer granularity is needed. This can be provided either based on the application or object level.

Application level means that information about users who use the same kind of application is provided. This is directly analogous to the COS level “who” functionality described above. Since the granularity that can be provided on application level is not fine enough and because this functionality can be realized as described above, this alternative will not be discussed further.

Awareness on the object level means that information can be provided if one or more other users are handling the same or a nearby object the user is operating on. The following issues must be investigated to provide awareness on object level:

1) How are user actions traced, i.e. how can we observe which (and how) objects are used?
2) How can we distinguish between different kinds of object usage and object retrieval?
3) How is the awareness information gathered and then distributed to users? What is the general model? Eventing, information on request?
4) What are the differences between awareness in synchronous and asynchronous interaction. How much of the information that was gathered for synchronous awareness can be transferred to use for asynchronous awareness?
5) How is awareness information provided to users? Which display techniques are needed?

These research questions are discussed in the following sections.

11.3.2. Trapping object access

Is there a general concept for object lookup, so that we can describe tracing in general terms? An approach is made in the following.

Before methods of an object can be activated it has to be retrieved from or identified in the object server. It must be retrieved if the methods cannot or should not be executed in the object server. If the object server is capable of executing object methods it still has to be identified. However, which of the two alternatives is applicable for a particular object system is not important for our considerations. First, we can generalize the alternatives for our purpose by an object-lookup operation and second this is normally transparent for the user. Therefore, in the following we will consider an abstract „object lookup operation" to describe the fact that an object is retrieved, activated, identified, etc. for further use.

11.4. Local Awareness

In this section, we examine the problems of local awareness. By this we mean the situation which arises when more than one user are accessing the same object; so that the awareness is focuses on a single object that the users are having a "rendezvous" on. As we shall see, however, there is more than one way for a user to access an object and therefore different levels of local awareness to be supported.

The object lookup operation described in the previous section provides a unique place where awareness information can be gathered. Whenever an object lookup operation is performed the identifier[1] of the object and the name of the user is collected. This can be part of an extended lookup operation. But gathering just this information for each and every object lookup will lead to a large amount of unspecific data which cannot be interpreted in a way that allows detailed awareness information.

For example: A user performs a pattern matching search on a large set of objects. When just the information is gathered that he did a lookup on all these objects, it is obvious that this user will get so much awareness information that this is no longer meaningful for him. Vice versa a lot of other users will suddenly see that this user is interested in the same objects as they are.

This example illustrates that not each lookup operation can be treated in the same way. We need a weighting for each operation that expresses its relevance for awareness calculations. Operations which potentially touch a large set of objects (e.g. search) are less weighted than an operation that points to exactly one object (e.g. read). The following mapping of operations to weighting factors is proposed:

Operation	**operation weighting factor**
read object by name for further use, e.g. for displaying and or modifying the whole or parts of the object	1.0
read object to use it as a referral, e.g. to handle it as an icon or entry in a selection list	0,7
search using a boolean expression	0,5
search using regular expressions, e.g. pattern matching	0,4

Table 11.1

The factor for search operations appears to be quite high but it is further refined by incorporating the number of retrieved objects per single operation (n) into the awareness factor calculations. The following formula is proposed:

A : awareness factor **=** weighting factor **x** \F(1;\R(16;n))

We use the square root of **n** rather than just simply **n** as we require a shallow curve of awareness rather than the sharp drop off in awareness that **n** would give us.

1 The technical term identifier is used here because it cannot be assumed that all objects in an object-system are named.

11.5. Spatial awareness in object systems

11.5.1. How far away is the next object?

So far we have only considered awareness issues for the case that users “meet” on the same object. But, even more interesting and important is the case where users operate on objects which are near to each other. Obviously, the term near is very vague and cannot be transfered into elements of the object model very easily. The following examples illustrate this.

Shared text / graphic editing: In this example paragraphs, sentences, letters, headings, etc. or graphical entities like lines, points, rectangles, etc. can be identified as objects. The most obvious measure for the distance between two objects is their distance within the whole text or graphic. Although, for both it cannot be measured in the same metrics. While in a graphic the distance can be measured in a length metric (e.g. cm), in a text it is more reasonable to measure the distance in paragraphs, sections, or chapters. So, in one example the distance is calculated by an external measure, i.e. their geometrical position, and in the other it is measured counting the number of the same or related other objects. Thus we have an example for an absolute and a relative measure. In both cases the distances is measured relative to an object that comprises the objects in question.

Rooms world: In a rooms world (such as those described in (Henderson 1985, Cook 1991, Leever 1992)) it can be said that objects are near which belong to the same room, building, or city? Building or city seem too large, but we have to consider the users focus, i.e. the viewpoints from which a user is observing the scene. We should be aware that these are not necessarily real world viewpoints. For example, in a virtual world a user might chose a viewpoint that provides him a view into and therefore awareness about a whole building. This is something that normally cannot be achieved in the real world.

However, in a rooms world, distance between objects can be measured using relationships between objects that represent their geographical location. In most cases the primary awareness of a user is focussed on objects which are located in the same room as he is. In this example the awareness focus can be easily calculated. But this raises the general question:

11.5.2. What defines the awareness focus[1] of a user?

For shared editing applications it is mainly defined by the view the user sees on his screen, although this can be enlarged by views that allow an overview.

But, let us consider another example where these questions cannot be answered in as straightforward a fashion as in the previous examples.

In a COS, there may be instances of objects which do not belong within a surrounding clustering object. Therefore other commonalities must be identified.

---

1 There is some similarity between this term and the terms "aura" and "focus" as used by [Benford 92].

These might be: type, class, shared values, relations between objects that can be used to derive a clustering object.

The following table summarizes the spatial relations that have been identified so far, and proposes a calculation method for the awareness factor:

Spatial relation	**relation weighting factor**
objects of the same type	1,0 x □□16;n
objects of the same class	1,0 x 1/(number of objects of this class)
objects of the same type hierarchy	suggestions welcome!
objects linked by a sequential relation, e.g. paragraphs of a text	relation factor **x** number of traversed relations
objects linked by a geographical relation, e.g. objects in a room	1,0 x 1/(number of objects in the room)^(1/16) x 1/(number of rooms between the object)
objects linked by any other relation *by assigning the relation factor 1,0 to the objects in room location this calculation method generalizes the geographical relation case. (to be deleted?)*	relation factor x 1/ (number of objects linked by that relation)^(1/16)
objects sharing attributes	suggestions welcome!
objects located in geometric dimensions (e.g. objects in a graphic)	calculate the distance between the objects

Table 11.2

11.5.3. Awareness Space

In general we can say: a relation with factor 1,0 is primarily used to describe “contained in” relations, i.e. a relation that groups or clusters objects together into an **awareness space**. An awareness space is a set of objects which are related to a clustering object. Not all objects in a room are of the same awareness to a user, but they will have similar awareness factors.

A clustering object is therefore an object that can be used to specify a user´s awareness focus. Thus, if a user is focussing on a room, the room object is the awareness focus, and the user will be aware of all objects which are in the awareness space of the object. How aware of the various objects he is depends on the awareness factor of the objects.

The following function includes the spatial factor into the awareness factor calculation.

B : awareness factor per object **=** operation weighting factor **x** relation weighting factor

In this paper, we have considered the following examples :

a) database browsing and querying

b) shared text and graphical editing

c) the rooms world

Table 2 lists relation weighting factors for various operations. The operations are chosen from the database viewpoint. Taking the examples discussed above additional operations must be discussed, since only the third example is a database like example. For the rooms example the following operations need to be weighted: enter a room (1,0), walking along a floor (0,2 for rooms) etc.

We are still considering how weighting factors can be defined for shared drawing/editing.

The following figure illustrates a scenario where two users are operating in an object system.

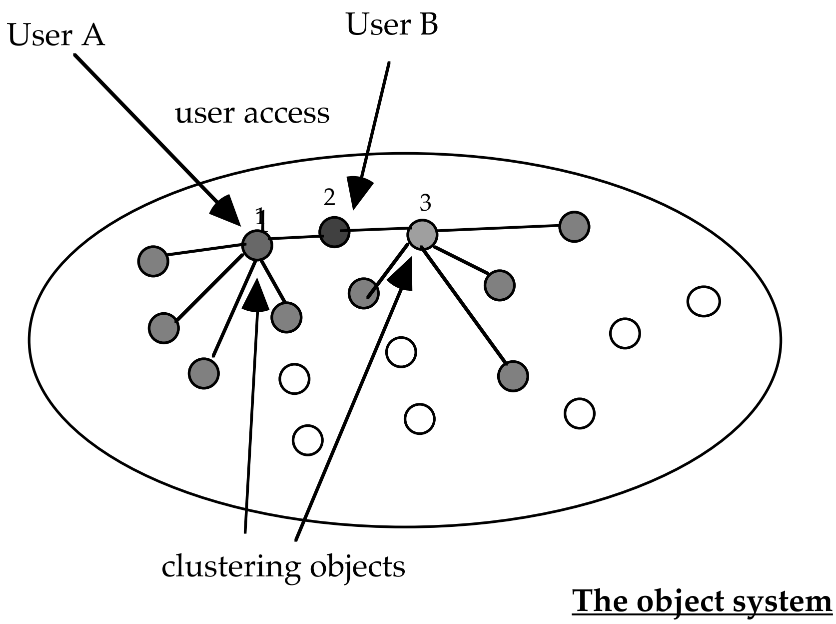


Figure 11.2

User A accesses object 1 which is a clustering object, lets say a room. So, his awareness focus is on object 1 and by that he is aware of all objects in the awareness space of object 1. User B access object 2 which is not a clustering object, lets assume a tool. This object is located in two different object clusters, e.g. rooms. Now, is the following questions must be answered.

- How aware is user B of the objects in the awareness spaces of object 1 and 3?
- How aware is user A of user B and vice versa?

Proposal: A user who is directly accessing non clustering object, has a reduced awareness factor (x 0,6 ?) on objects which belong to the awareness space of that object.

Not all objects are the same!

So far we have assumed that all objects are the same. The awareness factor was calculated without considering the object itself, i.e. only the operations by which it was access are discussed. But it is obvious that there are some objects we are more aware about than others, so we cannot treat them all equal. Consequently, each object has itself an awareness factor between 0 (simple data objects) and 1 (active objects like video displays, loudspeakers, white walls, etc) which is considered in the calculation:

C : awareness factor per object **=** operation weighting factor **x** relation weighting factor **x** awareness factor of object

The default awareness factor is specified for an object type and it can be modified for each instance individually.

Do we need different object awareness factors for different operations?

11.6. How aware are users of each other?

Until now, we have discussed the awareness factor for objects a user is accessing directly or indirectly (through an awareness space). How do we calculate the awareness of two users?

Approach: The awareness of two users is calculated by the product / sum of their awareness factors of those objects where both factors are greater zero:

```
FOR all objects
        WHERE awareness(user A) > 0 AND awareness(user B) > 0

mutual_awareness += (awareness(user A) + awareness(user B)) / 2
```

The problem with this calculation is that it is not normalized, i.e. its value can be greater than one. Another calculation alternative is to take only the maximum of a joint awareness object. But this does not cover the case where two user touch the same ten objects peripherally (factor 0,2). In that case they should be more aware of each other than 0,2. How can that be calculated? using a sqrt?

11.7. Awareness in Time

How do we handle the situation where a user has “touched” an object but some time has passed since he refered to that object? We need the concept of fading awareness. Possible solution : for each user, we keep a list of instructions they have executed. If it is N instructions since the user touched object A we calculate the user´s awareness factor of A as follows : (etc). Either this or there is an explicit “close” or “finished with” operation on an object that a user or their client must execute (i.e. in a COS browser, when the window is closed then the user is “finished with” that object). Another approach would be to timestamp each instruction and fade awareness out with the passage of real time.

How the history information is stored. How it is used, how it might be used?

11.8. Management and Evaluation of Awareness Information

The calculated awareness factor must be associated with the object and the user in order to provide awareness information to other users. How is that done? Possible solutions are:

a) stored with the object

b) stored with a user object

c) stored in a user/object matrix

Solution c) has the advantage that this information can easier distributed among user in order to update the awareness information.

General issues:

- the mapping list: operation — awareness factor must be represented
- it must be decided where the awareness information is stored. It seems that a list or matrix seems appropriate. The same distribution mechanism as with the user list can be applied.
- a stronger concept of operation transaction is needed for the calculation of the number of objects which are retrieved per operation.
- perhaps this is also needed: each application of the COS (in particular a Browser) must provide the semantic of an operation. This is then mapped onto the factor using the mapping table. Is that a too strong requirement?

11.9. Displaying awareness information

We must bear in mind that any such display of information should not disturb (or annoy) the user. Two views are possible:

- the awareness between each other and
- the awareness of user on an object.

The latter is important for objects which are active, i.e. which change their status or which give acoustic or visual signals.

We must consider where to display the awareness information. We might choose to display it with the object or in a separate display. If it is displayed with the object then we could argue that it is associated with the user´s current focus of attention. On the other hand, it might obscure the information which is actually part of the object. If it is displayed separately, then the user can choose to ignore it and use the system as if they were the sole user. They would only need to look at the awareness information if they were actively interested in it. Perhaps we could use a hybrid solution, where the awareness information can be popped-up on the object or perhaps a system-wide flag may be available, to switch the awareness information display on and off. In the above argument, we are assuming that a list of user names might appear alongside the object (or in a separate list of objects), and perhaps the type of access might also be displayed.

Turning to more graphical solutions, it might be nice to use a radar screen approach, with blips on the radar to represent users and their distance from the center to represent (weakening) awareness. This is a 2D approach, but we only have a sensible definition for one of the dimensions (distance from center). Perhaps the other dimension just needs to be generated in order to fit a number of blips on the screen. Another possibility would be to use the other dimension to represent the passage of time. Are there any three dimensional possibilities?

We could use a graph view of the current locale of the object store, with the objects appearing as nodes in the graph with users attached.

11.10. Remarks on privacy

Even if the provision of awareness information will proof useful for users, it becomes apparent that it involves a privacy problem since awareness can be provided only when user actions are traced and evaluated. The following list gives some arguments which try to weaken that aspect.

- Users who try to touch a large set of objects, in order to see where other users are working will have no success because of the weighting factors.
- WISIWYS: What I see is what you see. Whenever I´m aware of somebody he´s aware of me, so that I can see if somebody is tracing me. So there is information for some social control?

11.11. Summary

This chapter has presented an approach to provide awareness in object systems. The approach is based on the weighting of operations in respect to their relevance for cooperating users.

It is assumed that a lot of evaluation is required in order to get a good measure for the awareness factors.

There are still a number of open issues and decisions to be made. Nevertheless, it is intended that most of the mechanisms outlined will be prototyped in a particular example of a COS, namely the RADIO (Relationships And Data In Organisations) system. We expect feedback from this implementation will provide the evaluation required to settle on awareness factors and on HCI issues.

12. Mapping a knowledge sharing environment onto a shared object service

Yngve Sundblad

KTH

A mechanism for sharing knowledge between the participants in a distributed environment for collaboration is a “library” of expertise areas, i.e. references to “who-knows-what”-information, as parts of a *KnowledgeNet,* maintained by the participants themselves.

Here we study how some specific user situations in the KnowledgeNet environment can be mapped onto a shared object service, intended to provide the functionality needed for sharing objects in collaboration situations, including information on interesting events, awareness of other partners’ actions, query functions, etc.

12.1. Introduction and background

This chapter is a study of how a model on sharing knowledge in collaborative environments, the *KnowledgeNet*, (Marmolin & Sundblad 1993), can be mapped onto a *shared object service,* which follows the requirements given in (Mariani & al 1993), thus bringing together for synergy two activities in the Esprit Basic Research Action COMIC.

One important need in modern work life is efficient handling of information in a society that produces so much information that traditional text-based and TV-based media are insufficient. The need to handle the “information overflow” has been characterised as a change in the social paradigm of society (Kumon 1992) and different visionary computer based solutions have been suggested (e.g. Bush 1945, Nelson 1980, Engelbart 1990). These solutions are all focused on the management of published information in global and open but personalised libraries. The KnowledgeNet is another kind of solution, based on the sharing and integration of unpublished private knowledge using Computer Supported Cooperative Work, CSCW, technology.

The aim of the COMIC shared object service is to provide functionalities which allow objects to be shared by a community of users. The distinguishing feature from existing multiuser storage services is the focus on sharing and the provision of mechanisms which support the management of this sharing. It is intended that the shared object service abstracts from the properties of underlying infrastructure to provide a well defined abstract set of services. This offers the advantages of portability and allows the project to consider the services which need to be provided in terms of an abstract computational model which can then be realised on a number of platforms.

With its user interface, the collaborative desktop, CoDesk, the KnowledgeNet forms a general collaborative work environment. CoDesk, described in (Marmolin & al 1992a), is designed to support the management of collaboration in a manner similar to document and application handling in e.g. the popular Macintosh desktop interface and consists of interface objects representing members, catalogues, groups, documents, expertise areas, CSCW tools and private tools. CoDesk is based on a room metaphor, which models the computer network as a set of virtual rooms or spaces within which people interact. Mapping CoDesk onto the shared object server will not be the topic for this but for a separate study, for which the interaction and presentation parts of SOS need to further elaborated than in (Mariani & al 1993).

12.2. KnowledgeNet model and objects

The KnowledgeNet is designed to support a vision of the social work situation in which collaboration among peers can take place by sharing and integration of knowledge. The information overload in the society of today is often handled by using other people as references rather than excessive reading of documents (see e.g. Curtis & al 1988, Kedzierski 1988, Berlin & Jeffries 1992). With the KnowledgeNet we aim to support this process by shared knowledge bases of experts accessible by CSCW tools and make undocumented knowledge public in the same way as libraries make documented knowledge public.

There are many ways in which undocumented knowledge can be shared in a distributed environment. Firstly, knowledge can be broadcast in the form of lectures, announcements etc. Secondly, knowledge can be obtained by explicitly asking others for information and advice. Thirdly, knowledge can be implicitly transferred during meetings. Fourthly, a database about "who-knows-what" can be made accessible and maintainable by the participants. The KnowledgeNet aims at supporting all these ways of sharing and integrating knowledge by using appropriate CSCW technology.

The KnowledgeNet could be viewed as a distributed "library" of documented and undocumented knowledge that is made accessible by CSCW technology. Areas of undocumented knowledge are made public by expertise declarations describing the kind of knowledge the member possesses. The library consists of these expertise declarations and the documents produced by the members with links to the originator of the information in a way that supports communication and collaboration with appropriate CSCW tools. The information space is then peopled and the originator of the information is made visible and accessible. The library is user controlled, i.e. the members themselves decide if a document should be made public, define and maintain their own expertise areas etc. The most important concepts or objects of this design model are members, catalogues, groups, expertise areas, documents and CSCW tools, as illustrated in the prototype interface in figure 12.1, and described in more detail below.

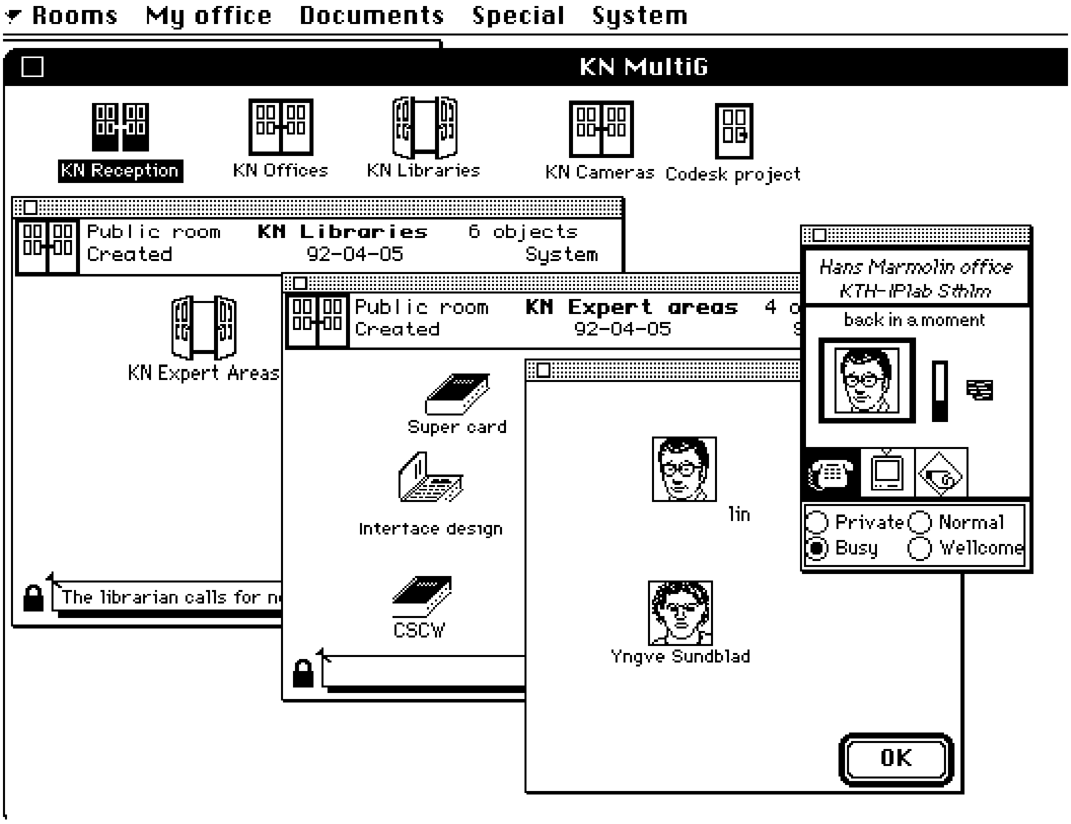


Figure 12.1 : An interface prototype (CoDesk) to the KnowledgeNet showing an open expertise library, with expert areas, experts and the office of an expert.

The information provided by the KnowledgeNet has to be kept up-to-date in some way. This is of crucial importance for the use of expertise areas, documents, catalogues and members. The updating can be handled in two ways, either by the members themselves or by a system administrator. We prefer the first solution in accordance with our emphasis on user control. However, the KnowledgeNet should provide different kinds of support for keeping the information updated. Examples of such support are registrations of how these information elements are used (how often, when and by whom), procedures for deleting and modifying unused elements, both own elements and other members elements, procedures for “subscribing” to changes., mechanisms for being aware of the presence of other users/experts. Some of these mechanisms can be provided by a shared object server as illustrated in section 4 of this paper.

12.2.1. Members

A member is any user that has enroled into the KnowledgeNet. The enrolment process has the form of a social contract in which the member promises to contribute to the knowledge in the KnowledgeNet. This is done by making expertise declarations, i.e. the members describe their areas of expertise or knowledge.

An important property of the members is their communicability status. In the KnowledgeNet we distinguish between members that are not present (not logged in), members that are available and could be reached by other members, members that are busy and not available for the moment and members that want to be completely private.

12.2.2. Catalogues

A catalogue is a container for information about the members of the KnowledgeNet., such as

- the "who-knows-what" catalogue that describes what kind of knowledge or expertise the members possess
- the "who-does-what" catalogue that describes the members commitments and what they are doing for the moment
- the "who-is-who" catalogue that describes the different members,
- the "frame-of-reference" catalogue describing important concepts, agreed goals, the politics of the information etc.

12.2.3. Groups

Members can create subgroups within the KnowledgeNet. Each group constitutes a context for collaboration and is defined in terms of the members, the shared documents and the CSCW tools used to accomplish a certain collaborative task. Groups in the KnowledgeNet could be more or less persistent. The most casual group is the group defined by the use of a certain CSCW tool. Such groups exists only when the tool is used. A more persistent group is a group created by a member as described above. It exists as long as the collaborative task exists. The most persistent group is the KnowledgeNet itself.

12.2.4. Expertise areas

An expertise area is an area of knowledge declared as such by one or more of the members. Each expertise area has links to the corresponding experts and support for communication and collaboration using appropriate CSCW tools. In addition there are links to related areas and documents. An expertise area is an area of competence in which a member perceives herself as having knowledge with which she can contribute to the knowledge in the net. Expertise areas are defined by the members themselves in terms of a keyword, a description of context and tasks and a degree of expertise, e.g. education, experience and produced reports. As an example members in our project define exprtise areas such as X.500, C++, ISIS, user interface design and communication. Also the links to other expertise areas and documents are defined by the members themselves.

One member, the *knowledge net librarian*, should be chosen as responsible for reviewing the proposed areas from the perspective of consistency and resolving conflicts between keywords. This will of course not guarantee that expertise areas have the same meaning for all members, nor that all agree that a certain member is an expert in some area. However, this is also the case in a traditional environment and it has to be solved in the same way, i.e by social learning, which may be supported by registering the communication pattern and the use of expertise areas.

12.2.5. Documents

Documents produced by the members could be public, shared or private. Public documents are stored in the library and could be borrowed by other members, who also could subscribe to changes, always getting the latest version. Public documents have the same kind of CSCW-links to the originator as expertise areas. There could also be hypertext links, created by the authors themselves, between documents and between expertise areas and documents.

12.2.6. CSCW tools

Basic CSCW tools in the KnowledgeNet are the synchronous and asynchronous communication tools used to access and cooperate with the originator of the information in the information space (picture phone, telephone, electronic mail etc.) and to distribute information (bulletin boards, “overhead projectors”, message systems, shared documents etc.). In addition more specific collaboration tools, such as co-editors, could be integrated into the KnowledgeNet if they meet the basic requirements for sharing knowledge.

12.3. Shared object service functionalities

The feature which most distinguishes the COMIC shared object service from other multiuser storage systems is the focus on sharing and the provision of mechanisms which support the management of this sharing with the explicit goal to provide an awareness of the action of others on objects within the shared object service. Thus, in the COMIC shared object service when an object is accessed or altered the systems should inform the relevant users the action has occurred in order to promote awareness across the service.

The overall structure of the shared object service is shown below in figure 12.2.

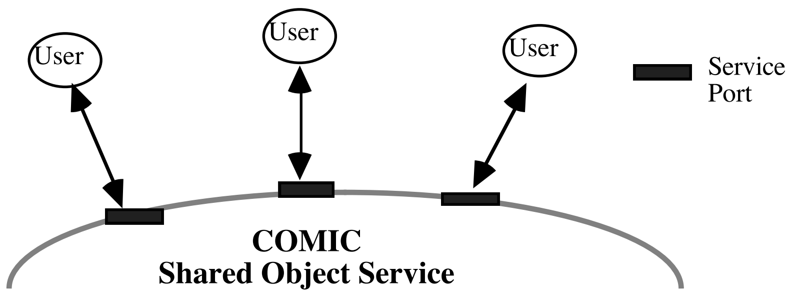


Figure 12.2: An overview of the shared object service

The shared object service is controlled through a set of commands issued to a service port which allows users to select the facilities they require.

Within the COMIC service objects have a number of significant components

- An object ID, which is a unique identifier associated with a single object and gives a means of access to that object
- A table of attribute value pairs, which represents the state of the object instance
- A type tag, which allows the object to be queried to find out its type
- A set of interfaces to the object, each containing contains a list of methods which can be invoked on an object and provide a means of accessing the object state

An object is as shown in figure 12.3.

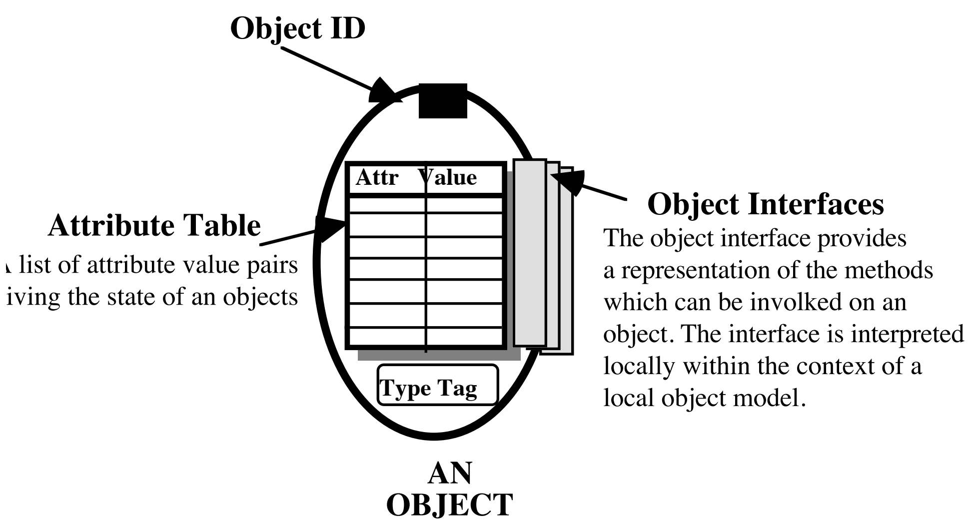


Figure 12.3: An object representation within the shared object service

This representation allows an object to be registered with the service to allow it to be shared by users of the service. In addition to the representation of objects, the shared object service needs to provide a set of mechanisms to allow the sharing of objects to be managed by the service. Some of these services, that are of specific interest in KnowledgeNet situations, are described in the following subsections.

12.3.1. Events

The COMIC shared object service should enable users of the shared object service to be aware of the actions of other users of the services. In addition, the shared object service should provide facilities to allow users of the shared object service to make others aware of actions of interest to them.

The mechanism used to provide this form of awareness within the service is events. When any action on an object that is of possible interest for other objects takes place events are generated. The shared object service provides facilities to allow events to be defined, created, related to other users and handled in appropriate ways. Users and objects can declare interests in certain events or types of events, see the section on subsciption below.

Within the COMIC shared object system events belong to a specially dedicated class of system objects. These object as in the case of all other objects within the service are shared across the community of co-operating users. The distinction is that a set of specific services are provided to handle event objects. These services instruct the shared object service to inform the user when specific actions occur relating to event objects.

12.3.2. Subscriptions

The Subscription Handler of the object server is kernel to the services that support users' (and objects') awareness of each others' actions. This tool should support the clients with services needed to contract, or later retract, certain events or types of events. An object announces an interest in other objects or events by declaring an interest to the Subscription Handler or directly to the objects of interest.

A client, application or user, could declare interest in a certain event in the system, internal or external, by declaring the interest to the Subscription Handler. If the subscription is an external entity then a "shadow", or proxy, object that represents the real object is used.

Mechanisms are needed for, e.g., a subscriber's

- immediate information about changes in an object
- information about changes in an object when accessing it
- immediate information about anyone taking interest in (access to, or even just pointing at) an object
- information about someone else having accessed an object, when accessing it
- information about interests of other users, as manifested in subscriptions

With these mechanisms a subscriber can promote her awareness of other users of the collaborative environment but mechanisms are also needed for giving a user awareness of others without having to subscribe.

12.3.3. Awareness

In some situations it is useful to have an overview of what other people are working with, regardless of the representation on the screen. An example is someone studying my recent work that is not completely ready yet. I want to know that because it is probably good that I talk to her directly and explain it.

If the situation is that people are talking to each other, there is obviously a need for them to be able to talk about object presented to them on their screens, but if the object are presented differently it may be hard to know that it is the same object one is talking about. One solution to the problem is to use a highlight operation on objects in the interface and then distribute it, so that the object is highlighted in all applications regardless of view used.

A key question is to understand what information needs to be transferred between applications to make the persons able to work together. Information that represents the status of the application and needs to be transferred includes:

- information about operations performed by the other person
- information about which objects she is focusing on
- information about what objects she is manipulating

We also need to transfer information that is not bound to any operation on the data but is used to allow persons to talk about objects they are manipulating. Operation here include highlighting an object or a group of objects and mark objects in some way.

One of the key concepts identified early in the COMIC research has been the issue of awareness. Described simply, users need to be aware of each others presence and activity within the system in order to initiate collaboration. There have been (initially) two approaches towards this topic; one, the object approach (Mariani & Prinz 1993) considers objects within an information store (typically a database system) and users browsing or generally accessing the objects within the store. The spatial approach (Benford, Fahlen, Bowers) considers a Virtual Reality setting.

The object approach uses a variety of formulae to calculate an awareness factor between two users; this takes into account both “space” (how near are two objects to each other : here, again, various formulae and object classifications are used to provide this spatial metric) and “time” (how long ago was a user accessing a certain object: this is used to provide the fading of awareness over time).

Based on the information held in the event objects, the service agent (the Awareness Manager) can calculate awareness factors between objects and thence between users (of those objects). Until now, it would seem that there is little that needs to be added other than concern over the attributes of the event objects. However, it must be taken into account that not all objects are created equally.

Some objects may possess a greater “pay attention to me” attribute than others. We suggest that every object possesses a PATM attribute and accordingly methods to get and set the value of that attribute.

12.3.4. Queries and views

Data objects can be handled differently by different application, for example experts in different fields may wish to look at an object in different perspectives suited for their domain. When various views on an object are used there is also often a need to have alternative ways to manipulate the object and the operation performed may be of interest to the people one is working with.

It is important to be able to retrieve objects by a query specification. Within the shared object service a query specification will be modelled by an object. Such an object will consist of a list of object "facets" against which will be a set of required values and an arithmetic relational operator (i.e. equal, not equal, less than, etc.). Facets will also be related by Boolean operators (and, or, not). Such a query specification (with the matching "query engine") should give us the computational equivalent of general ad-hoc query mechanisms within standard database systems.

With database query mechanisms, there is a sense of the operation being carried out "for once and for all". In other words, the answer (collection of objects matching the query specification) is valid for the current state (or snapshot) of the object store. Within the shared object service the result of a query is itself an object. If we query for all objects in a collection that have a certain value of a certain attribute we get a collection object whose state is represented by a set of ObjectIDs. We can keep that collection ID around for a while, long after the query was first issued. If the shared object service supports persistency (which it should), then we can return to the collection at some time in the future. In this case, what happens to the collection if a new object with the same attribute value is added to the base collection that the query operation was carried out on in the first place?

There is thus a need to specify views as transitory (the resultant collection is discarded as soon as the view operation that brought it into being is out of scope of the surrounding transaction) or persistent (the resultant collection is saved and updated if new objects are added which match the view condition). To support the dynamic addition (and indeed, deletion if some object changes the value of their first name attribute) of objects to a collection, we need some sense of a "future" query, which can be handled by the event mechanism.

12.3.5. History

An important kind of query that has to be supported is on history of objects. The objects must contain time stamps on creation (and deletion when applicable). In order to keep the granularity of time specification on a per-object basis we assume that such an important change of an object that the previous state should be part of the history creates a new object with reference to its ancestor.

Given these assumptions, the "Query Engine" should be able to provide history if it is constructed to cope with temporal clauses delimiting the range of a query to a set time period.

12.3.6. External references

It is important that “real physical” objects can be subject to similar kinds of operations as the objects stored within the shared object service. For example, just as we can issue queries regarding stored objects, it would be nice if we could issue queries regarding real physical objects such as “Who currently has a copy of a specific book?”.

It is also important to realise that objects stored within the shared object service may make reference to “real physical” objects every bit as much as they may make references to other objects stored within the SOS.

In some senses, an object which contains a reference to an external object can be considered as being incomplete. Unlike a fully internal object, we do not possess all the information “at hand” to display, manipulate, indeed generally consider the object. This impacts on issues of how do we display or present an incomplete object, how to we query and retrieve an incomplete object, etc.

How do we choose to represent these external references? One way might be to actually have a “pseudo” or “proxy” object” which is actually stored within the shared object service but which simply contains details about the “real physical” object. A problem is that different users might enter the same external object with somewhat different descriptions.

A possible answer to this kind of problems is to admit that incomplete objects will always be incomplete! The purpose of the external reference is merely to give us a hook into the physical objects, where presently none exists. If we are searching for a copy of the specific book and we can query the (incomplete) objects which reference the book we can hope that we may find some clues as to whose office currently contains it. Event and history considerations can also assist us; we may be able to find out who recently has been searching for the proceedings (just as we are) and perhaps that person was successful in her search and might be a good starting place.

12.4. Mapping some knowledge sharing situations

Here we describe how some knowledge sharing situations can be mapped onto the shared object services described above. In the situations we strive to maintain as much user control as possible, resolving conflicts in competing events by asking the user which to take care of first.

12.4.1. Looking for expertise

A KnowledgeNet member searching for anyone knowing about a certain subject starts by looking at the expertise areas, which are objects with names that can be subject to *queries*. There might be a perfect hit when the query word exists in the expertise library, but quite often there will be a need for ingenuity in choosing similar or related terms. The query mechanisms should allow for search on stems of words, alternatives etc.

A hit gives a list of (icons) of persons, some of which or all of which can be opened communication links to at the choice of the searcher. The hit is an *event* that is sent to those chosen by the searcher and/or to those having *subscribed* on being contacted at every, or this certain kind of, hit on the actual expertise area. The event results in a demand for contact via the CSCW tools chosen by the searcher or available by default.

Depending on the communicability status of the expert member contacted (not present, available, busy or not to be disturbed) the event result is direct communication, a knock on the door or pending for, i.e. subscribing for the event of, the member's presence. The seracher can also choose to send an asynchronous message, an email.

In the communication between the searcher and the expert there might be documents (objects) related to the expertise area that the communicators should share. Sharing or looking for *external objects* might also be of vital interest. These objects might be seen as subscribers to the interest in the area, activated when such interest occurs.

Awareness of other members of the KnowledgeNet communicating on the same expertise area or a related one or on a related document can of course also be of great value.

In cases where the area looked for is not found or the expert is not available it might be of value to contact the knowlege net librarian, whose icon should be available for sending messages or direct communication and who should be sent alerting events on anomalies. Mechanisms for viewing *history* of searches and contacts made can also be of great help.

12.4.2. Expert entering

When a member enters (logs in to) the KnowledgeNet, or becomes available from having been busy, an event is created. That event is checked towards any subscriptions, of which demands for contact from other members are likely to be common. The entering member is notified on which these pending contact demands are and, especially when they come from expertise area search, on what subject. She then chooses if and how to try to contact the waiting members and whether to inform some of them that they will be contacted in due time.

When deciding when and how to make contact, features of section 4.1, such as awareness and history, can of course be of interest here too.

12.4.3. Changing expertise area

Membes having shown interest in a certain expertise area might by default or actively have been registered as subscribers of changes in the description of that ara or even in a certain person's description of her own knowledge in the area.

When a member enters or changes her expertise description for a certain area an event should be created and distributed to all subscribers.

The introduction of a new expertise area or canceling of an area or of membership in an area should result in a contact with the librarian and an event sent out to those subscribing to such major changes. When stating a new area the member can be asked about related areas (prsented with a list of areas) and te subscribers of the related aras should also be notified by the event distributed.

12.4.4. Librarian intervention

The knowledge net librarian has to moderate and intervene when expertise areas become overlapping or confusing. All definitions of new or retracting of expertise areas should thus lad to events sent to the librarian with a possibility for direct contact. Of special value for the librarian is views of history of area definitions, searches and member contacts.

12.4.5. External expertise

External “objects” of special interest in the KnowledgeNet application are external experts, rpresented by internal objects with information about their competence and on how to contact them.

The mechanisms for handling them can be very similar to the handling of internal expetise, as described in section 12.4.1.

12.5. Conclusions and further work

Here we show that the basic mechanisms in the shared object servic under developmnt within the COMIC projct are directly useful in situations encountered in the KnowledgeNet environment.

An important part of a collaboration environment is the user interface, which has greta implications on the usefulness of the computer support. The crucial interaction and presentation parts of the shared object service are under development and will be used in a similar study as this of how the KnowlegeNet interface, i.e. CoDesk, the collaborative desktop, can be mapped onto the service.

Prototypes of the share object service will be developed for different platforms within COMIC. Building future versions of KnowledgeNet, for which prototypes have already been developed, on the shared object mechanisms will be considered.

13. COLA: A lightweight activitiy platform based on shared objects

Jonathan Trevor, Tom Rodden, and Gordon Blair

University of Lancaster

Despite the reliance of cooperative applications on the facilities provided by distributed systems, little consideration is given by these systems to the support of cooperative work. This chapter examines the provision of appropriate mechanisms to represent cooperative work within a distributed platform. Based upon a examination of existing models of cooperative activity and the experiences of their use, a lightweight model of activities is suggested as the basis for the supporting platform. Rather than concentrate on the exchange of information, this lightweight model focus on the mechanisms of sharing of objects. This focus enables a clear separation between the mechanisms provided by the distributed platform and the policy which is the responsibility of the cooperative applications.

13.1. Introduction

The technologies currently exploited to construct cooperative applications were designed and developed prior to the emergence of CSCW and its associated applications. The facilities provided by these systems and the manner in which they are presented seldom sit easily with cooperative applications (Rodden, 1992). The needs of the application domain, the nature of the work and the mechanisms to support cooperation are intertwined within the development process. The developers of CSCW systems are forced to juggle all these factors in an attempt to realise a cooperative system. This is a painfully slow and problematic task resulting in a set of similar services replicated across a collection of applications in a manner that is confusing to both developers and users alike.

It our belief that a key requirement for developers of CSCW applications is the need for specific support for the development of these applications. Applications programmers should be free to concentrate on the semantics of the application and what should be provided. The programmer should not need to worry about how the mechanisms employed obtain the required result. The development of this support is essential to the future of CSCW and we would agree with (Patterson, 1991) when he argues:-

> “If multi-user applications are to flourish in the future, then programmers will require support for building these applications” .

This chapter presents a platform to support group working by providing mechanisms for sharing Cooperating Objects in Lightweight Activities (COLA). The COLA platform provides the means to allow applications to externalise appropriate features of cooperative activities in such a way that these can be shared across applications. Previous activity models have adopted a modelling perspective based on communication and its influences on work flow or document flow. In contrast, our approach is based on mediating the sharing of information rather than controlling its exchange. Consequently, the emphasis is on the provision of suitable supporting mechanisms within the platform. This approach allow the details of policy and the associated control of this policy to be administered by the applications being supported.

13.2. The Nature of Support

The need for support has been addressed within computing in a variety of ways relevant to CSCW. At the lowest level, existing distributed systems provide support for many features of cooperative applications, for example, object migration and location transparency. In addition, some distributed system designers are considering more advanced features which could prove useful to CSCW. These include multimedia objects, group multicasting and high performance networks.

Distributed systems are far removed from cooperative applications which embody a set of assumptions of why people work together and often characterise how they should work. These applications contain considerable information about the domain in which they are applied. A number of CSCW *application toolkits* have emerged to support application development in different cooperative domains. These toolkits provide a set of existing facilities for a particular domain which can be reused by developers in the construction of new applications. Examples of this form include RENDEZVOUS (Patterson et al., 1990) and LIZA (Gibbs, 1989) which support the development of multi-user interfaces and DISTEDIT (Knister and Prakash, 1990) a development toolkit to support the construction of shared editing systems.

These toolkits provide little or no support for representing the cooperation taking place. However, a number of *cooperative environments* exist which focus on representing cooperative work and how these representations can support the work taking place. Tremendous variety exists within these environments and a number of different classes can be identified. Some indication of the classess and the diversity of cooperative environments embrace are shown in figure 13.1 .

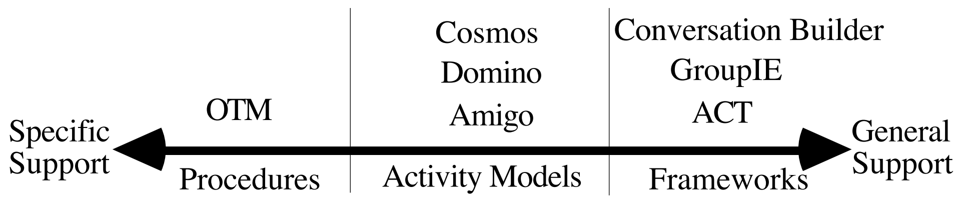


Figure 13.1. Spectrum of cooperative environments

In general, cooperative environments tend to be *goal oriented*; in that they include some conception of an activity or task which has some goal marking its completion. The more specific the support, the more specialised and narrow the goals. The majority of these systems are both computational and procedural in nature, However, *conceptual frameworks* exist where support is limited to the provision of models, outlines and frameworks upon which more specific instances can be built.

Procedural models describe well defined and understood tasks, almost exclusively in an office environment. The models control who takes part in a task, what operations they should perform in order for a task to be completed and what documents need to be exchanged. The model enforces what the user does and "runs" the procedure. Examples include OTM (Office Task Manager) (Lochovsky et al., 1988) and the systems currently being developed by action technologies based upon the COORDINATOR(Medina-Mora et al., 1992).

Activity models expand the horizons of procedural models by presenting a more general approach to task specification. In a similar manner to the procedure models, activity models have specific goals for the cooperation they are supporting. The main difference is that these models are general enough to be applicable to more than just those specific goals. The models concentrate on what and how information is exchanged between members of the activity and attach more significance to what these people can actually do. Many different systems offer these features, some of the most well known being COSMOS (Dollimore and Wilbur, 1991), DOMINO-W (Kreifelts and Woetzel, 1987) and the AMIGO ACTIVITY MODEL (Danielson and Pankoke-Babatz, 1988).

Frameworks are the most general form of cooperative environment, they aim to provide support for cooperation, but without any specific domain in mind. The ACT (Activity CoordinaTion) model (Kreifelts et al., 1991) supports the coordination of activities in groups or teams. The ACT model differs from activity models (such as AMIGO and COSMOS which focus on coordinating the communication flow) by concentrating on coordinating the execution of actions. Other examples adopting particular perspectives on group work include GROUPIE (Rudebusch, 1991), CONVERSATION BUILDER (Kaplan et al., 1992) and OVAL (Malone et al., 1992).

13.2.1. Shortcomings of Existing Support

All applications, interfaces and tools rely, to a greater or lesser extent, on the underlying services provided. However, any problems with this support will inevitably be propagated upwards when the service is used. This is one of the major reasons why CSCW applications are so difficult to build. Not only do the applications contain many issues single user applications do not, but support can be unstable and often unsuitable for the demands that are placed on them.

The most obvious drawback of existing cooperative environments is that they are closed applications rather than open platforms and do not provide facilities to support a number of different approaches to cooperative work. In addition, the particular problems of merely extending or augmenting existing support models include:

i) Unrealistic Models of the real-world

One of the biggest shortcomings of existing support for cooperation is that they start from a set of unrealistic assumptions about work. For example, OM-1 (Ishii and Ohkubo, 1991) and other activity models represent well structured cooperative work knowledge. These assume that the information necessary for a task is know in advance and that the work follows a set procedure. In reality, work is not well structured or defined (i.e. the handbook is not followed and procedures are used as a resource rather than merely interpreted (Suchman, 1983)). Ishii and Ohkubo (1991) have also found this to be true in their experience of office tasks:

> " we found that office workers made many short-cuts and modifications to the standard procedures defined in the handbook. Therefore, it was no easy task to determine the actual standard procedure, even when it was defined clearly in the handbook".

Given that cooperative applications are intented to support the actual work of groups, unrealistic assumptions about that work will have tremendous impact upon the success of CSCW systems. Many authors have commented on the variability of work within natural settings and the difficulty of modelling this; interested readers are referred to (Bannon and Schmidt, 1992) for a full discussion of the issues involved. To minimise these problems we wish to reduce the set of assumptions within our support platform to the minimal set necessary to support cooperation.

ii) Constraining models of control

One of the main stumbling blocks of many activity models are that they rely on people behaving methodically and working to some plan. However, by constraining their actions users are being restricted by the model intented to help them. In fact, users often circumvent procedures and do the unexpected (Schmidt, 1991). This variability needs to be reflected in the support structure to allow applications to cope with variance from the expected norm. The handling of procedural exceptions in existing approaches is symptomatic of the problems of control. If exceptions are allowed, they can at best only be handled in a very general way. This leads to prescriptive models, which eliminate the possibility of exceptions and increase the burden upon the user.

Services and support tools should assist and augment higher levels of abstraction, and not prescribe particular viewpoints. The purpose of a support platform is to aid both the programmer and user, not to force them to do things in a particular manner. Consequently, a platform needs to provide the *mechanisms* necessary to represent cooperation. Policy surrounding control and coordination remains the exclusive responsibility of users or where appropriate applications.

iiii) Lack awareness of "groups"

With the possible exception of a few CSCW environments, existing computer platforms, tools and services provide only limited awareness of others, thus users are unaware of who is cooperating on what. It is often the case that people gain new insights and ideas from others and there is no reason why this shouldn't be the case when people cooperate together on a computer. A supporting platform should provide a high level of "*group awareness*", with users aware of the actions of others.

iv) Limited support for "sharing"

Cooperative work relies on people *sharing* information (ideas, files, etc.). However, the majority of activity models attempt to coordinate people in cooperative work through a model of cooperation based on asynchronous message passing (for example, forms based systems). Little support is provided for the sharing of objects to support the cooperation taking place.

We would therefore, characterise much of the existing support as heavyweight with a high degree of application specific semantics encoded, and enforced, by a model based on message passing. We wish to consider the development of a lightweight model which adopts a perspective to cooperation based on sharing.

13.3. A Lightweight Approach

Future CSCW applications will desire a great deal more than current support can provide. Using the problems above as a basis, we can identify three major desirable characteristics for cooperative support platforms:

i) A lightweight and flexible representation of cooperation

ii) A separation of the application semantics from the support features

iii) The provision of increased group awareness

Our approach is to design a platform which directly addresses these requirements. This will allow additional services to develop more realistic models of cooperation and allow further study into suitable and realistic underlying support for cooperation. In essence, this lightweight approach needs to provide useful *mechanisms* for describing cooperative situations, while relying only on *minimal semantic knowledge* in order to interpret these mechanisms.

Our focus is on the representation of cooperation which augments existing distributed services and communication systems. The COLA (Cooperating Objects in Lightweight Activities) platform provides a lightweight service which *aids users and applications in the cooperative use of objects*. The central part of the platform is a lightweight activity model which is used to provide a context in which objects can be shared. Unlike many cooperative environments, the bias of this platform is towards providing mechanisms to support sharing but with limited semantics. This means the platform acts as a "veneer" between semantically laden cooperative environments and distributed systems.

COLA presents mechanisms to cooperative applications and environments building upon the general mechanisms provided by distributed systems. With this approach:

- the only semantic information put into the activity are the events that change the state of the activity and the stages that an activity goes through.
- an activity can move forward and backwards or jump to any stage. The lightweight model does not specify when this occurs.
- objects can be shared amongst activities, and can be accessed from many different contexts (even from outside any activities)
- objects are context dependent and can present different interfaces to different people at different times.
- every change in an activity produces an event. Events are delivered to anyone who has specified some interest in it. Therefore, anyone in the system can be as aware as they like.
- users can move between activities as they wish and may undertake some roles local to an activity.

Activity *policy* rests exclusively with the application — the enforcement of deadlines and other features of activity control are contained within the application. The COLA platform can be considered as providing a set of *policy free mechanisms* to allow different features of activities to be represented. With a clean mechanism and policy separation the platform not only allows existing models to be built on top of it, but enables users to circumvent applications and directly interact with the platform.

The platform provides two important interfaces. The first is through a defined service interface available to cooperative applications. This allows applications to register activities and update the information within the activity model using remote procedure calls. An equally important component is the lightweight activity browser which provides a user interface to the activity model which allows activity information to be directly accessed and manipulated by users.

13.4. Components of a Lightweight Model

The lightweight model adopted within the COLA platform focuses on providing mechanisms to allow the externalised nature of activities to be represented and shared. Consequently, the platform is less concerned with either the structure or intent of the cooperation taking place but more with representing external effects of the cooperation. A number of distinct components exist within the lightweight model.

- *Activities*, provide a structure for the cooperation.
- *Roles*, limit object access and presentation in an activity.
- *Objects*, which are unaware of their context and*Object Adapters*, which present objects in context sensitive manner.
- *Events*, enable any objects registered in the platform to be kept group aware.

This section explains these main aspects of the platform in the following sections, for each, a brief example is given to illustrate its use.

13.4.1. Activities

A lightweight activity is defined as *a process in which users and objects interact and exhibit a public state*. A number of people participate in activities and an activity has a *life-cycle*, subdivided into *stages,* which it moves through before completion. Advancement and retreating of stages in an activity is done by the applications involved in the activity through the activity service provided by the platform. There are several reasons why the conditions required for a change of stage are not recorded in the definition of the activity (the "activity template"):

- this information is the sort of semantic information which may not be known in advance
- exceptions may arise outside of the activity which means a stage may need to be skipped or retraced.
- often, even simple events are not fixed and are open to negotiation within a cooperative setting.

As an example, consider the setting of an examination associated with an undergraduate course. The question paper should be created and approved before the examination takes place. The paper may be submitted for approval and edited any number of times. The exam commences at a given time and lasts a set duration. Answer papers cannot be created before the exam starts, after the exam is completed no further additions to the answer papers can be made.

In this scenario, heavyweight activity models would focus on the construction of this activity and its decomposition into constituent subtasks. A traditional activity model would attempt to capture the dependencies between sub tasks, the deadlines associated with different tasks and the behaviour exhibited by different roles. In contrast, the lightweight model within the COLA platform focuses on the external effects of this activity by defining only the stages the activity must move through and the objects and people associated with it. The stages to an examination setting activity could be described as:

CREATEPAPER STAGE

Purpose	To create an exam paper for students to complete
People involved	Writers (of the paper), Reader (to check the paper)
Objects	Exam paper and Sample Solutions

DOPAPER STAGE

Purpose	For students taking the course to read and complete an answer paper using the exam paper previously created
Roles involved	Student (of the course), Examiner (to solve problems with the paper)
Objects	Exam paper and Answer sheets.

MARKPAPER STAGE

Purpose	To mark all the answer papers produced by the students
Roles involved	Marker
Ends	Answer sheets and Sample Solutions

13.4.2. Roles

In an activity, people do different things to contribute to the work taking place. For example, within the examination activity their exists people who:

- set the exam paper.
- check/proof read the paper to make sure it is satisfactory.
- answer the questions
- mark the answer papers

In order to represent this variability, people can take on different *roles*. Some roles may be taken by more than one person, some may only be assumed by one. Previous activity models have used roles to explicitly describe the behaviour of different people, in contrast, roles are used within the lightweight model as a means of access control. This access control is sufficient to represent within the framework the profiles of the different people taking part in an activity. That is not to say that the application using the activity model cannot *prescribe* activity related actions or conditions on that role but this is not within the platform model which wishes to remain semantically neutral.

Within the examination activity a number of distinct roles can be identified:-

WRITER	The role concerned with writing and creating the question paper
READER	This role makes sure the paper is satisfactory.
STUDENT	The users taking this role create and write the answers during the exam.
MARKER	Anyone in this role marks the students answer papers.

Many of these roles may be occupied by the same person (e.g. in reality the writer of an examination paper and the marker are often the same) and at each stage any roles are free to participate. However, the roles within an activity need to be encoded in such a manner as to highlight illegal[1] access operations, such as a student reading the exam paper before the exam starts.

13.4.3. Objects

Objects present the largest problem in a lightweight activity model. They can be accessed in several ways, from within an activity where they were created, from inside another activity or from outside of any activity. In each case the object may present a slightly different interface to reflect the invoking users rights. Objects in a lightweight activity model may also present a different set of operations (an interface) to each role during an activity. Thus an object exhibits a degree of *activity sensitivity*. However the basic object itself may still be activity unaware, but is merely presented in different ways to different roles and at different stages of the activity.

Accessing an object from within the context of an activity, through a role, means that the object must respond differently depending on the accessing role and the stage an activity is in. Accessing an object from inside another activity requires more rules for different roles for the second activity. More importantly the security of the object could be easily compromised as two activities may have no knowledge of the restrictions each has placed on the objects. Clearly it is cumbersome in the extreme, if not impossible, to define huge sets of rules on each object operation for every potential eventuality. The solution adopted within the COLA platform is the use of *Object Adapters*. which provide a clear interface between the objects within the platform and the entities which can access them.

Object Adapters

Object adapters have attempted to capture aspects of both object presentation and access control, and as such, have much in common with many methods in both of the presentation and access areas (such as capabilities and proxy objects (Shapiro, 1986)). However, object adaptors in COLA embody a set of concepts concerning the cooperative sharing of the object. Every object has a basic set of operations. When an object is created from an activity that activity is given an object adapter that abstracts the basic object into a set of activity meaningful operations, together with a set of access rules for that object. The object adapter is initially defined as part of the activity and is stored, along with existing pre-defined activity templates (such as an exam activity) in an information store.

When an object is used, an instance of an object adapter is created which as far as the user within an activity is aware acts as the object and presents an appropriate interface to the them (as illustrated in figure 13.2). The object adapter provides several operations beyond normal interface filtering:

- *Extended operational semantics:* These methods masquerade as standard object interface operations insofar as the client would view the object interface

1 Although appearing illegal, access might not necessarily be invalid. For example, it could be a matter of policy that dyslexic students are allow to read exam scripts in advance.

without the adapter. However these new methods have deliberate side effects which change the underlying object into an object which is contextually sensitive. e.g. a normal “write” operation on a file will look the same to a client as the “write” operation on the underlying object, but will also perform some locking information.

- *Dynamic Contextual Filtering*: Unlike a normal interface to an object, which cannot change once it has been assigned, an object adapter provides a set of rules based on an associated activity’s state and the client’s current status within the system. These rules dictate what operations a user can invoke at any time during the activities life cycle.
- *Extendible Interfaces*: These are new operations specially provided by the adapter which use and act upon the semantic knowledge of what the object is supposed to represent with the activity.
- *Interface amalgamation*: No assumption is made that an adapter only presents one object interface. This means that an adapter can present a combination of more than one interface (and hence more than one object) to a client. The client only sees one object through the adapter

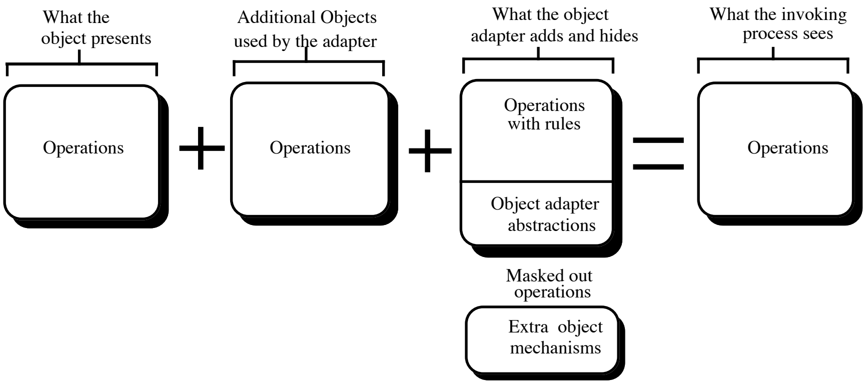


Figure 13.2. Presentation of an object

Each set of basic adapters and objects are managed and controlled by a*master* adapter, or simply a master. In effect the master adapter distinguishes the objects owner. This “ownership” can be transferred to any other activity or user at any time.

The rules describing which operations are usable by who, and when, inside the basic adapter and master are defined using a combination of the rules based upon the activity context, the role of the client, and the clients status in the overall system. If the object is *activity dependent*, e.g. an exam paper, then the set of allowable operations in the basic adapter will be very small. If the object is highly *activity independent,* e.g. a shared whiteboard application, then this set of operations may be large. One basic adapter is created per client of an object in order to allow the adapters to customise themselves towards particular clients.

It is up to the original activity, through the master, to promote clients to allow access to higher privilage operations. For each combination of user, role and activity which may access the adapter there is a corresponding rule and *access level*. Levels are a less cumbersome way of defining which operations can be performed on an object. Instead of listing the operations with each potential access, they are listed one (or more times) in the adapter and are ordered into one or more "*ladders*" of importance. Therefore the greater the degree of access you have, the "higher" up on a particular ladder you are, and the more operations you can perform. When the client invokes an operation on the object, through the adapter, the first rule which can be applied to the user (starting at the most specific) is applied. If no rules are applicable then the invocation fails. It is up to the original activity, through the master, to promote users and other activities to use higher privileged operations.

From the examination example, we can identify three necessary objects; the question paper, the sample solutions and the answer scripts. The following tables define the question paper as an abstraction of a text file which, for example, can only be read by someone in a student role when the activity is in the "DoPaper" stage.

UNDERLYING OBJECT: TEXT FILE

OBJECT OPERATION LADDER: 1

Level	Operation(s)
1	Write
2	Append
3	Read

QUESTION PAPER RULE SET:

User	Role	Activity	Required condition	Level	Ladder
any[1]	Student	*local*	stage = DoPaper	3	1
any	Reader	*local*	stage = CreatePaper	2	1
any	Writer	*local*	*none*	1	1
any	*any*	*any*	stage>= MarkPaper	3	1

13.4.4. Events

Events are small structured messages that are propagated around the environment in order to make environment objects aware of any changes of state. Events which the user or application are interested in and can generate are kept and managed by the platform. The awareness that events provide is primarily achieved by allowing users to register *interests* in certain types of event with the platform. When an event that matches this "interest" arrives at an activity it is delivered to the user (as well as the actual specified destinations). Events addressed to one activity from another, or events which are generated locally within an activity, are always seen by members of the activity.

An event is delivered according to three levels of addressing, the most general being to all the people registered to an activity. The next level is the role within an

1 Keywords in italics have special meanings. "any" in the user column for instance matches any user.

activity and finally the user themselves. It is not necessary to always specify the activity if a role is specified, nor a role or activity if a user is specified.

Events themselves all contain several standard fields. Each event must be *named* uniquely (within an activity) and contain a *description* that outlines what the purpose of the event is. Two addressing fields are used, the *source* and *destination*. Each of the addresses can be decomposed into *user* (which may be an object ID and therefore does not always refer to a human user), a *role* and an *activity*. The final part of an event is the *contents* field which can contain any number of text subfields which can have activities, roles and users tagged onto them to restricted reading and writing.

Any activity will have access to a core set of events. These include event which are commonly used between objects (e.g. for synchronisation) and by the user. One such event is the *notify* event, a very general message which can be passed between users, objects and a mixture of the two. Most events are a specialisation of this event. Within the examination example there are a few possible specialisations, for example:

PAPERREADY? EVENT

Purpose	To ask the READER role to check that the question paper is suitable of the exam.
Contents fields	The location / name of the question paper object
S'rce restrictions	Writer role
Dest' restrictions	Reader role

PAPERREADYREPLYEVENT

Purpose	To confirm to the READER that the paper is satisfactory or needs further work
Contents fields	Approval field and a text field indicating what work needs doing
S'rce restrictions	Reader role
Dest' restrictions	Writer role

EXAMSTART EVENT

Purpose	To inform the students taking the course that the exam is about to start
Contents fields	Examination start time and name of object holding the examination paper
S'rce restrictions	Reader, Writer and Marker roles
Dest' restrictions	-

COLA Events resemble *semi-structured messages* (Malone et al., 1986) (Ishii and Ohkubo, 1991) in many ways. These are text messages with identifiable types. Each type contains a known set of fields, but with some fields containing unstructured text or other information. The difference between events and semi-structured messages is in approach. Semi-structured messages concentrate more on the semantic nature of the information being sent in the message and its destination(s) whereas events are simpler. In effect, semi-structured messages can be viewed as a semantic extension and specialisation of events with events in COLA are closer to the use of events in Kronika (Lövstrand, 1991).

13.5. Functional Architecture and Services

The COLA platform, a number of simple applications and a *platform browser* which allows direct user access are currently being developed beyond an existing prototype. The prototype contains a fully working event model and a set of skeleton support services, coupled with a number of initial browser interfaces. ISIS (Birman and Marzullo, 1989) was chosen as a suitable toolkit to support the development in a distributed environment because of its use of process "groups" and associated multicast facilities. We want to compare and contrast ISIS as an underlying distributed support system with others, such as the ANSA TESTBENCH (ANSA, 1989) and CHORUS (Rozier et al., 1990), in order to address two concerns:

- what are the distributed systems requirements of CSCW applications?
- what aspects of existing systems are useful in supporting these requirements?

Figure 13.3 shows the various objects that make up the functional architecture (circles) and the interprocess communication streams used to link them (lines and arrows). The most important conceptual aspects are controlled by the existence of corresponding managers. There is one activity manager and event manager per activity and an overall object manager, called an object trader.

Each *activity manager* maintains an activity template, which describes the associated components of the activity, and acts as the central point of coordination and reference for the activity. The activity manager responsiblities include membership, role allocation and activity update.

The *event manager* provides event propagation to both activity members and other event managers as well as providing facilities to allow event interests to be registered and filters on these event registers. In the examination example suppose a user wished to see what events were going from Joe to Thomas, specifically in the activity known as "Examination course 123". The user would make up an event template with only the destination activity, destination user and source user fields completed (with the values "Examination course 123", "Thomas", "Joe" respectively) and then register this with *any* event manager, as all events are seen by every activity.

The *object trader* is split into two parts, the *finder* (where all objects in the COLA platform are registered) and *binder*. In order to obtain access to an object a user must put in a request or query to the binder (which take the form of an object template and a client context). The binder is the main point of access control and attaches a user to an object adapter to the underlying object.

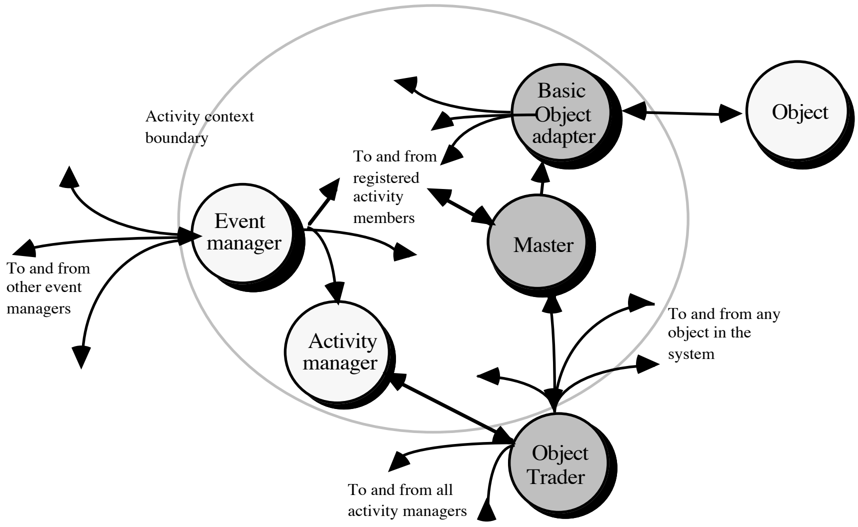


Figure 13.3. IPC calls between objects

13.6. Conclusion

This chapter has highlighted the need to provide appropriate support for CSCW applications and has indicated that existing support is inadequate because of a number of problems. A platform to address these problems, COLA, has been described which acts as a bridge between distributed systems (which provide many mechanisms for sharing but without any knowledge of *why* people work together) and cooperative environments (which contain a great deal of semantic information, and often prescribe *how* people ought to work together). It consists of a flexible lightweight activity model, roles for access control, events for group awareness and object adapters, which enable objects to be context sensitive.

The platform is unconcerned with the semantics of what happens when information is passed around, and provides the means to share the information, relying on either social protocols or further computer management to ensure the information is used appropriately. This approach is still relatively new, but it is clear that if the wide range of future CSCW applications are to become adopted as useful everyday tools, then lightweight support is essential.

It remains to be seen how much of the CSCW application developers overhead has actually been removed by the COLA platform and how much, in terms of performance, the developer will have to pay in order to reap the possible gains that COLA provides.

14. Supporting Lightweight Activities in a Virtual Environment

Jonathan Trevor

University of Lancaster

This chapter takes the spatial model proposed in Chapter Four, which has little support for planned or structured activities in which conversations may take place, as a starting point on top of which the concepts of the COLA model may be visualised in a virtual environment. It is hoped that this visualisation can combine the interaction structuring of COLA and the spontaneity of conversation in the Spatial model to produce an environment which remains highly flexible and dynamic in supporting intuitive object communication and interaction, while overlaying a variable degree of control and management for this communication and interaction.

14.1. Introduction

This project (Comic 1993), and Benford and Fahlen (Benford and Fahlen 1993) present a spatial model of group interaction in virtual environments. The model aims to provide flexible support for managing conversations among large groups gathered in virtual spaces. This support enables conversations between users to be ad-hoc and opportunistic. The lightweight COLA model (Trevor et al. 1993), and associated platform, provides flexible lightweight support for the structuring of group processes. This was accomplished by externalising the state of a group process, called an *activity,* and through the provision of group mechanisms with minimal semantic knowledge. The COLA approach gives little support for how the members of these processes actually communicate effectively *within* the activity.

Both these projects identified *awareness* as a key requirement for supporting collaborating groups of users. Within the COLA model, users are kept aware of the state of any activities in which they participate through events and an application independent system browser. This awareness is predominantly between *users and the cooperative environments* they are in, relying on other tools and interfaces for effective inter-user communication. Conversely, the spatial model focuses exclusively on awareness *between objects* in space (Chapter 6, Comic 1993).

This chapter takes the spatial model proposed in Chapter Four, which has little support for planned or structured activities in which conversations may take place, as a starting point on top of which the concepts of the COLA model may be visualised in a virtual environment. It is hoped that this visualisation can combine the interaction structuring of COLA and the spontaneity of conversation in the Spatial model to produce an environment which remains highly flexible and dynamic in supporting intuitive object communication and interaction, while overlaying a variable degree of control and management for this communication and interaction. In order to do this, the report concentrates on:

- examining how the central concepts within the COLA platform can be spatially visualised within a virtual environment using the ideas and terminology used in the Spatial model
- demonstrating how objects and users actually use and interact with the different aspects of this visualisation of COLA in space
- outlining a suitable, but general, architecture to support visualised activities. Because this project was used to help develop this visualisation, an architecture developed to be general may also be suitable for supporting the Spatial model.
- examine how dVS, an architecture specifically oriented towards building virtual worlds, could be used to actually present the ideas and services described above.

Virtual environments generate many new ideas (and associated problems) because of the large number of possibilities, concepts and shear depth in the field. These range from the presentation of a world in the environment through a three dimensional user interface, to the underlying network which supports it. Therefore, it is stressed that while this report attempts to answer the questions posed above in detail, in many cases it is only possible to scratch the surface and should be considered as a starting point for new ideas and possibilities, rather than providing exhaustive solutions.

14.2. COLA model

The COLA platform (Trevor at al. 1993) provides a lightweight service which aids users and applications in the cooperative use of objects. The central part of the platform is a lightweight activity model which is used to represent the cooperation as well as providing a context in which objects can be shared. Users are kept aware of each others actions, and those of the applications involved in the activity, by event delivery. Unlike many cooperative environments, the bias of this platform is towards providing mechanisms to support sharing but with few of the semantics. This means the platform acts as a “veneer” between semantically laden cooperative environments and distributed systems

Activity *policy* rests exclusively with the application — the enforcement of deadlines and other features of activity control are contained within the application. The COLA platform can be considered as providing a set of *policy free mechanisms* to allow different features of activities to be represented. With a clean mechanism and policy separation the platform not only allows existing models to be built on top of it, but enables users to circumvent applications and directly interact with the platform.

This lightweight model is adopted within the COLA platform, which focuses on providing mechanisms to allow the externalised nature of activities to be represented and shared. Consequently, the platform is less concerned with either the structure or intent of the cooperation taking place but more with representing external effects of the cooperation. A number of distinct components exist to provide the lightweight features:

- *Activities*
  A lightweight activity provides a structure for any cooperation, defined as *a process in which users and objects interact and exhibit a public state*. A number of people participate in activities and an activity has a *life-cycle*, subdivided into *stages,* which it moves through before completion.
- *Roles*
  In an activity, people do different things to contribute to the work taking place. To represent this variability, people can take on different *roles*. Previous activity models have used roles to explicitly describe the behaviour of different people, in contrast, roles are used within the lightweight model as a means to limit object access and presentation in an activity
- *Objects*, which are unaware of their context and *Object Adapters*, which present objects in context sensitive manner.
- *Events*
  Events are small structured messages that are propagated around the platform in order to make objects immediately aware of any changes of state.

Some of the features of the lightweight approach in COLA are:

- users can move between activities as they wish
- the only semantic information put into the activity are the events that change the state of the activity and the stages that an activity goes through.
- an activity can move forward and backwards or jump to any stage. The lightweight model does not specify when this occurs.
- objects can be shared amongst activities, and can be accessed from many different contexts (even from outside any activities)
- objects are context dependent and can present different interfaces to different people at different times.
- every change in an activity produces an event. Events are delivered to anyone who has specified some interest in it. Therefore, anyone in the system can be as aware as they like.
- People may undertake some roles local to an activity. These different roles represent different abilities within an activity.

The COLA platform addresses many problems of control found in CSCW, including the requirements for sharing objects, through cooperation oriented mechanisms. By enabling both the current state of any activity, and exactly what is occurring within it, to be externalised, COLA enables applications and users to be made *aware* of the cooperation in which they are taking part. This cooperation based state enables access control and presentation of object interfaces to be based upon the cooperation in which the object is placed, the knowledge of precisely who is using it, what data they are accessing and why they are doing so. The provision of object adapters allows objects to be have their operations extended and augmented to take this externalised activity information into account, while still avoiding embedding the semantics associated with that cooperation into the underlying object.

14.3. A spatial model for a virtual environment

Chapter Four (Comic 1993) and Benford and Fahlen (Benford and Fahlen 1993) define a virtual environment as a set of spaces, populated by objects, through which people move. The largest difference between virtual environments and other types of environment is that a spatial metric can be applied to the objects in it. The correct use of these spatial metrics of objects can provide far more advanced forms of conversation management between objects, in contrast with other approaches, such as floor control and workflow techniques. These existing approaches are too rigid and unnatural to effectively manager interactions between large numbers of people in a virtual space. The spatial model of interaction takes advantage of the highly fluid and dynamic nature of space in order to overcome these problems.

Although the model considers 3D space, the ideas are presented in a generic manner and can apply to anything where a spatial metric can be applied, such as higher dimensional Information Terrain's (the visualisation of information). The model has four objectives: ensuring individual autonomy; maintaining a power balance between "speakers" and "listeners"; minimising hard-wired constraints; and support for free mingling with the addition of formal mechanisms later. To achieve these objectives a number of key abstractions are necessary:

- *Medium*
  Interaction between objects occurs through a medium. Mediums may be audio, video, text etc.
- *Aura*
  Aura is subspace, not necessarily contiguous, which bounds the presence of an object within a given medium and which acts as an enabler of potential interaction. Different mediums have difference auras. The use of aura enables the environment to "bound" the amount of information that objects receive and operate on, therefore preventing cognitive and computational overload.
- *Awareness*
  A quantifiable level that measures how "aware" one object is of another. This need not be symmetrical between objects.
- *Focus*
  Focus is a managed space projected by an object, the semantics of which can be described as *"the more an object is within your focus the more aware you are of it"*

- *Nimbus*
  Nimbus is the counterpart of focus, a space where *"the more an object is within your nimbus the more aware it is of you"*.

These abstractions can be directly related to many models of cooperative work, where nimbus is a subspace in which one projects ones activities, focus is a subspace in which one attends to the activity of some entity and an entities aura is a subspace in which it projects its presence. The users themselves in a virtual environment do not need not be explicitly aware of using these abstractions as they may be implied through the users actions, use of eyes, direction etc. The project identifies three primary ways of users manipulating aura, focus and nimbus:

- Implicitly through movement and orientation
- Explicitly through a few key parameters
- Implicitly by using adapter objects which modify aura (e.g. a microphone object amplifies your audio aura and nimbus

Within the actual virtual environment aura, focus and nimbus are modified by objects in one of two categories, adapter objects which amplify or attenuate them and boundary objects which also effect traversal of the space. Boundary objects can be obstructive, non-obstructive, conditionally obstructive and transformational[1].

An initial three dimensional implementation of the aura, focus and nimbus abstractions was accomplished using the DIVE (Fahlen et al. 1993) system. Another system which exploits this spatial metaphor is CyCo (Benford et al. 1993b). This is limited to two dimensional windows where conversations are managed using focus and nimbus projected from flat representations of users. In CyCo three different levels of awareness between these users are available: fully aware; unaware; and partially aware, which means you know someone is trying to speak with you but not what they are trying to say.

A more formal computational framework for the spatial model can be achieved by relating it to the object oriented approach to modelling in distributed systems, such as ODP object model (ISO 1991). In order to do this, human beings are represented as objects, spatial objects are translated onto distributed objects, and the communications media is mapped to different interfaces for the objects. Unfortunately, the existing trading model (ISO 1991b) is insufficient to represent the ad-hoc and opportunistic methods of interaction that these human "objects" employ. Further, passive trading, based on waiting for the objects to invoke the trading service, is inadequate in a large virtual environment to support the interactions between the spatial objects because the interfaces involved are simply not known in advance. The objects now need to be actively informed when an interface is available to be used. Therefore, the Comic model presents an active trading service based upon managing object interaction through inter-object awareness. This awareness depends on the spatial proximity of objects involved (e.g. the closer you are the more you see of a bulletin board)

---

1 As indicated by the Comic authors, these objects seem to only be a special subset of adapter objects

An active trader is responsible for importing and exporting references to interfaces in the space and object auras. The collision of auras (and hence the potential for interaction) is detected using dedicated collision manager processes which are spawned whenever two interfaces of the same medium are present in the same space. When a collision does occur, the positions, angles and IDs of the objects involved are exchanged, which are then used by the objects themselves to calculate awareness levels based upon their relative focus and nimbus. These awareness levels between the objects are used in two ways to control the subsequent interaction between them by:

- operations being associated with an awareness threshold at which they become available to other objects
- objects deciding to invoke operations on others interfaces once certain thresholds are passed

The Spatial model provides mechanisms for managing conversations between people and objects in spatial settings which are controlled using the awareness. Aura, focus, nimbus and adapter objects, coupled with the spatial properties of the objects (e.g. position and orientation), all contribute to this notion of awareness. The model can be applied to any number of spatial settings, two of which (virtual reality and text conferencing) are being currently prototyped.

14.4. Virtual Concepts and Tools

The presentation of the COLA model in a spatially oriented manner relies on a combination of the Comic spatial model and a number of general concepts and tools. These concepts and tools are introduced below in order to provide all of the necessary framework in which the visualisation of lightweight activities can take place.

14.4.1. Reactive managed spaces

Collision between nimbus and focus spaces produced an asymmetrical awareness level between the objects which the space originated from and enabled interaction between these objects to occur. We view these spaces as specific instances of *reactive managed spaces*. Several features of the spaces can be identified. They:

- react when collision occurs with another space or object
- carry private and public information
- do not have to be linked to an object, they can simply “exist”
- do not have to be continuous. e.g. they can cover several objects which can be in different areas of the virtual environment.
- may have a finite life span after which they “die”
- each have an associated management object which controls the above features.

In the most general instance, a collision of spaces and objects will produce a predefined sequence of actions based upon the identity of the colliding space and the relative spatial orientations of the objects and spaces involved. These actions are defined when the manager is created but may be modified during the managers lifetime. Where managers are attached to objects (who *project* the space), the space itself will perform little processing. It is generally the responsibility of the owning object to determine what actions to take.

Consider realising nimbus and focus spaces as reactive managed spaces. The managers for the spaces will be programmed to only respond to a collision with other focus and nimbus spaces. Collisions will inevitably occur with other types of spaces and objects but no action is taken by the focus and nimbus spaces. This lack of action is not necessarily mutual, the other "uninteresting" colliding spaces themselves may react. Once the space does collide with a focus/nimbus space then both of the spatial managers involved attempt to fire a sequence of actions/code associated with any collision with a space of that type. A focus and nimbus collision causes the managers to calculate an awareness level of the object projecting the other space based upon the current spatial coordinates of the collision. This awareness level is then subsequently passed to the object owning the space which decides what to do. In this case, the awareness level causes a variable number of operations to be presented to the colliding object.

14.4.2. Virtual adapters and Visors

Underwater divers wear flippers, breathing equipment and a wet suit in order that they can effectively move through, and interact with, their environment. For the same reasons cameras, and other tools a diver uses, are also specially protected to enable them to be used underwater. We draw the analogy between objects in a virtual world and those in an underwater environment. We argue that while the user needs a headset (etc.) to actually "enter" the environment, *successful interaction* with the virtual environment requires their virtual body/object to be *appropriately extendible* .

Therefore the most general tool which any virtual object requires is a *virtual adapter*. This object performs two functions, insulating an object from the virtual environment and facilitating the objects interaction with the virtual world. It presents standard interfaces to the object it surrounds; the collection of objects which are attached to the "adapter" object and the other objects in the virtual environment. These surrounding objects include other objects with virtual adapters and collections of reactive spaces. In general, virtual adapters connect with other tools which aid the underlying object in interacting with their environment and have "pockets" where these tools may be stored when not in use. These inactive tools and space objects can therefore be carried around with the adapter, and are immediately available when the user wishes to "wear" them. The virtual adapter acts as a sort of "personal object database" which can be accessed wherever the object is.

A user in any large virtual environment with a large number of objects, and a large number of reactive spaces, is faced with a daunting amount of information at their disposal. This information must be successfully sorted and filtered or the object is faced with computational and information overload. We introduce a portable and general tool called a *visor*. Visors are the tangible means through which users can see and interact with managed spaces. By “wearing” a combination of visors (i.e. connecting them to the virtual adapters surrounding the them) objects can detect as many, or as few, different spaces as they wish. Visors “work” by projecting their own reactive space which reacts when it collides with other spaces and objects. The actual consequence of this reaction, and the types of space which produce it, are determined by the visor.

To illustrate these tools consider the following example. Steve enters a virtual environment. The object representing his body has an associated virtual adapter which carries several default visors; three nimbus and three focus. These cover some basic spectrum of modes of conversations that Steve may want to use. Initially, without any visor, Steve has a representation in the environment, but no means to interact with other objects. By selecting a wide focus space and a wide nimbus space Steve shows he is interested in everything happening around him. Once he is engaged in some conversation, through the collision of nimbus and focus spaces, it may be necessary to restrict his nimbus (by either editing a current visor or selecting a new one) in order to avoid other people listening in or distracting him!

14.4.3. Remote controls and gateways

Remote controls can connect to virtual adapters and act as front-end interfaces, or browsers, to underlying applications which must travel with the user. The advantage of modelling interfaces in terms of portable “remote controls” are twofold:

- users do not need to physically travel to a special space which has all the application information and interface laid out ready for browsing and manipulating.
- applications which are moved to a virtual environment with existing flat interfaces can be used immediately, perhaps prior to development of a spatially oriented interface

Section 14.5.4. uses remote controls to encapsulate the method of generating *gateways*. A gateway is a link between two disparate spaces which the user enters in order to travel between them. In addition to connecting two points in normal space, gateways may link to very different spaces, such as *information terrains*, where data can be displayed and manipulated in more than two dimensions. Both of these destinations are used by the visualisation activity model. It seems useful to capture any alien features to a users general expectation of space (since you cannot “leap” between points in the real world[1]) within familiar concepts and tools to the user. In the real world, remote controls do have non-visual, abilities which often cannot be provided without using the control (a car alarm for example). While putting gateways “inside” of remote controls hides the concept inside of an established tool, the actual social and physical protocols of using gateways may need to

1 ...yet!

be considered in more detail. For example, someone literally disappearing in the middle of a conversation and an instant later arriving in the middle (physically and temporally) of a virtual meeting may produce some surprise to all the users involved, not to mention the possible disorientation the actual teleported person may feel.

14.5. Using COLA in a virtual environment

This section considers how the main features of the COLA model can be visualised, presented and manipulated within a spatial (or virtual) environment. The actual underlying support for this realisation remains a variant or extension of the underlying COLA platform. The basic concepts remain the same at the underlying platform level, although intermediate and new services may be required. A majority of the spatial concepts to aid interaction and bridge the gap between the virtual environment and the COLA platform come from the Comic model, and the same terminology is used to wherever possible.

14.5.1. Activities and externalised state

Visualising Activities

Initially, the most attractive idea is to represent an activity as a room that people wishing to participate in the activity walk into. Anything (person or object) within that room is therefore encapsulated by, and talking part in, the rooms activity. However, this visualisation assumes that an activity can be mapped to a single contiguous space and has no meaning outside of the room. Within a virtual environment, as with the COLA platform and real world, people and activities are not necessarily restricted to a specific space. People should be able to move between activities as they wish, and the objects belonging to an activity can be dispersed between many areas in the same space or room, and indeed, may also change space completely.

Our approach is to model activities in terms of reactive managed active spaces which surround any activity owned and shared objects. This means whenever an object moves, the space around that object moves with it, allowing activity objects to be fully dispersed while still maintaining the linked to the activity. Note that this does not preclude the intuitive solution of representing an activity an a room, since a room can be modelled as an object.

A consequence of allowing activity objects and members to be dispersed within the virtual environment, an activity user may need to differentiate between the various objects in any particular space, and be able to find out quickly what objects belong to activities which they are members of. This is accomplished by a user wearing one or more *activity visors*, selected from their virtual adapter object. These visors are “given” to the user when they join an activity, together with a default role object (see 14.5.3). A visor “sees” specific activity objects in the environment by projecting a reactive activity managed space of their own which responds with similar activity spaces. These activity spaces around objects come from the role object that the object wears in order to participate in the activity. When a visor activity space reacts some indication is given to the visor owner that the object belongs to the activity (e.g. it flashes, the name of the activity becomes visible, or even you can see the hats that each object may have). This "reaction" between spaces enables users to locate and differentiate between a large number of objects in a space very quickly.

Interacting with activities

While visors can detect objects and people involved in an activity, the next step that needs to be taken is to present the current activity state to the user within a virtual environment in some readily available and recognisable form. This remains an important feature of the COLA model and is even more crucial in an environment where the activity can be thoroughly dispersed.

In COLA, every activity is essentially an instance of a defining activity template. Therefore, the straightforward solution is to use an activity based remote control object. All the control needs to do is display the information contained within an activity template in some readable form e.g. lists of members, the roles they are playing, the objects which have been created etc. Users can bow see the externalised state of an activity "at a glance". Only one activity remote control is ever required by any acting member because every activity template can be used to obtain the same sort of information. The control chooses which template to display based upon the current role badge the object is "wearing". What makes the remote control tool more powerful than a passive display which simply shows up to date information is the inclusion of an two dimensional interface (with buttons, menus etc.) which the user can interact with. By making the control more interactive, it can now be used to browse and manipulate the COLA platform to create new activities, new objects in the current activity, join and leave activities etc. Therefore an activity remote control is very closely related to the original COLA browser.

14.5.2. Events

Visualising Events

Events are crucial to lightweight activities in allowing people to kept aware of any changes within the activity which may affect them. For example, the movement of an examination activity from the stage where the question paper is being approved, to the stage where the paper is made available to all the students (who will sit the exam) could cause an **activity** event to all students telling them that the paper is now ready. Events are also important at a **system** level to allow information to be exchanged between objects and processes.

The concept of an event at a virtual environment level is different to the activity and system approaches. For example, the destruction of an object in an activity may cause an event. The notification of that event can either be accomplished through a message being delivered to everyone who is interested in the object, through an activity event, *or* in an subtlety different manner to a *nearby* objects, through noise. This method of notification is impossible in any other environment since no spatial orientation can readily be applied. Therefore, we classify several different types of events which can occur in a virtual environment supporting lightweight activities:

System events

These are the underlying messages passed between objects at the distributed system level. They have no semantic attachments and only carry data from one point to one or more destination objects (identified through a system level address). These types of events cannot be visualised or shown since their contents have little meaning outside of the objects sending and receiving them. These are dealt with by the underlying system and will not be discussed further in this section.

Activity events

In the COLA platform events are sent between activities and objects within those activities. These events provide an essential cooperative awareness between the activity objects in the system. Event delivery uses three levels of addressing: user; role and activity. Very general events can be sent to everyone (just by filling in the activity field) or specific messages can be sent between people, possibly regardless of activity, by filling in the user field. Activity events do not travel through space, nor would it make sense to model them as such

Virtual events

Within a virtual environment, events are physical objects with a finite life span which when generated may be seen and travel in space, to and from any other objects. As is the case with other objects in the virtual world, events can have associated managed spaces. These spaces reflect the **consequences** of the event. Therefore, a virtual event is defined in terms of:

- the number and nature of the spaces which will surround the event

  Since the event is an object, similar to any other in the environment, it can have associated spaces. These spaces model the "effects" or consequences of the event itself. For example, one space may be a audio space which carries an "explosion" noise, indicating someone has destroyed an object.
- the description of the events spatial traversal

  A virtual event is an object which, like any other object, can travel within space. The ability to travel in space means that virtual events may be delivered in a number of ways:
  - to a set of spatial coordinates, i.e. to a point of virtual space
  - along a set of spatial coordinates, e.g. the event may travel in a straight line until a collision occurs.
  - no traversal, the event does not travel anywhere.
  - to a specific set of objects, specified in terms of an object template
- a terminating condition

  This determines how long the event will last before it gets destroyed. This final addition can be specified in a two ways:

  - a lifespan, some time after which it ceases to exist, e.g. 5 minutes
  - a specified number of certain collisions e.g. after colliding with three focuses
- the underlying object

  This is an object like any other in the virtual environment and may present different interfaces and data according to the spaces the object is colliding with.

Sending and Detecting / Receiving events

Event templates in the COLA platform could be registered to allow users to be as interested (or disinterested) in **any** of the events occurring in **any** of the activities in the system. The ability to change the amount of information coming into an object (or user) prevented computational and information overload as well as maintain a variable degree of activity awareness. It is equally important to provide an equivalent service for virtual objects who may receive too many virtual events to cope with.

Activity events

In the virtual environment, a user sends activity events in a similar manner as they would using COLA browser, using the activity control pad to select and fill in event templates. The events are then redirected by some COLA event manager towards the relevant object destinations.

Activity events are delivered to the objects in question by the activity event manager. The registration of event templates as interesting and disinteresting occurs at this manager which then filters each incoming event (to the activity) using this information. Once an event arrives at the object it is displayed on the activity remote control which informs the user, perhaps using a “paging” audio beep.

Virtual events

Virtual events are represented as a normal objects within a virtual environment and can be generated in the same way as any other object, using some environment object factory service or through some “virtual event remote control”. Once the parameters for an event template have been filled in (lifespan, contents, destination etc.) by the user, or a previously created event fetched from a database, the event can be created (by some factory service). It then automatically travels and reacts according to the specified parameters.

To receiving or detect travelling spatial events, objects must attach an “event detector” object to themselves (probably through a virtual adapter). The event detector projects an event space (figure 14.1) which reacts when events and event spaces (the consequences of an event) collide. The underlying user or object can use the detector in same manner as COLA objects use the event manager to register templates of events they wish to know about or ignore. On collision the detector interacts with the event to extract information and give this to the user (or perform the “consequences” of the event space collision) or ignores it depending on the recorded event interest/interest templates.

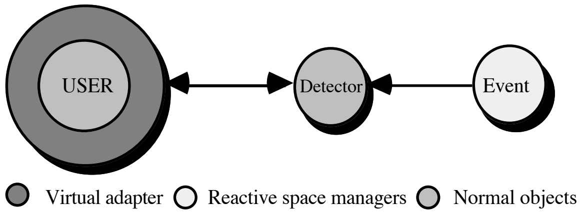


Figure 14.1: Event detection

Example of activity and virtual events

In an *submit-paper* activity, two members, Tom and Steve, have roles *creator* and *reader* respectively. The activity has three stages: creation (of the paper); checking (that the paper is ready for submission) and submission (the completion of the activity). Tom writes the paper during the creation stage and then uses his activity remote control to advance the activity to the next stage and causes an activity event to notify the *reader* role that the paper is ready for checking and the papers location. He then leaves the space. Steve arrives and checks the paper through. Once Steve has annotated the paper he moves the activity back to the creation stage. The activity automatically sends out a change of state activity event to all members. Tom is a long way from the space with the paper and therefore uses a gateway to directly transport himself to the paper. His arrival causes a virtual event. This event is centred on his location and does not move and has a lifespan of 5 seconds. It projects a consequence space across the immediate surroundings. This space then collides with Steves event detectors event space and informs Steve that Tom has just appeared behind him (through a noise or other medium). Steve and Tom then discuss the changes that need to be made.

14.5.3. Roles and activity context

Visualising Roles

COLA roles are used as a means of access control to shared activity object operations. More than one role can be assumed by any one person or object and are presented along with the persons identifier and current activity as credentials whenever the object wishes to perform an action on a shared object. Moving the idea of playing roles to a virtual environment causes two problems: how to present the role information to the shared object the object wishes to interact with; and how to show other members of the activity what role(s) the object has.

In reality, roles, such as supervisor, manager, team leader, technician etc., often do not need to be explicitly determined or made public (to other members of the activity) but are simply "known" facts. For example, a student will know who their supervisor is just by looking at them, and therefore the role that person has ("supervisor") i.e. their shape is sufficient to be linked to the role. In this manner the concept of a role, in general, is directly linked to the visualisation of that person, and therefore unique 3D personas are very important.

In a virtual environment we augment a persons 3D shape in order to provide additional visual clues about the roles that person is playing. At least two methods of doing this can be identified:

- Different hats for different roles

  When a person adopts a role they are given a hat. The more roles they have the more hats they are given. In order to display more than one role at once (other than stacking hats!) we could use the "Mad Hatter" approach, in recognition of a certain hat wearing animal (Carroll, L. 1865). These hats are different from the normal hat because they each have a label sticking out (figure 14.2) which may be read by others and names the roles that person is playing in the activity. If multiple roles are assumed by that person, the label gets "longer" in order to list the roles (and number of hats!) that person has, and the role player only "wears" the hat corresponding to the role that they are currently using. This solution provides a very immediate visual role identifier but relies both on a detectable difference between hats and knowledge of what role corresponds to what hat.

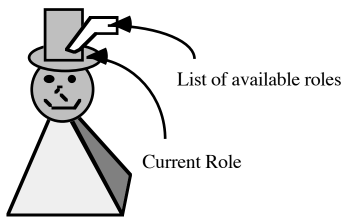


Figure 14.2: A "mad hatter"

- Identity badges

  Identity badges have the advantage that they can easily describe who the person is and what roles they are playing by having his information printed on the badge. In a similar manner to adding ribbons to badges at conferences, *coloured* badges can be used to distinguish roles at a distance. Further, different shaped badges, such as stars, rectangles and triangles, could differentiate between activities.

The problem with both of these approaches is scalability. With hats, there are only a certain number of possible shapes, and with badges a finite number of colours which can be distinguished at a distance. We choose the badge approach, while less immediate in its visual impact than hats, as they can more naturally display information about the user as well as the roles the person is playing.

Changing context: activities and current role

An objects "context" describes the state the object has within COLA model, and is used to control actions the user can and should perform within the activity and on shared objects. COLA expresses this context as a triple consisting the object/users name, the role(s) it is playing and the activity in which the user is currently participating in. In the original browser a users context automatically changed as they moved through activities. This movement depended on which activity interface they were manipulating. The current role was changed using a button press over a list of possible roles. Translating this method to a virtual environment, while still preserving the old method (using the activity remote control), enables a new, more tactile, approach to changing context — using badges. *A user or object is said to be active within an activity if they are wearing a role badge which belongs to that activity*. A user may be a non-active member of other activities simultaneously by wearing other badges *under* the currently active activity (and badge). By choosing different badges to be worn as the uppermost over of all the activity role badges, objects can move to different activities. The badge object itself exerts a managed space over the bearer, which informs other objects coming into contact with the object of its current activity and role — its context.

Example

Steve joins an teaching activity and is automatically assigned a blue role badge for the activity with a default role. Steve interacts with an interface to the role badges he is wearing to raise the teaching activity badge to the top, thereby making himself active in that activity using the default role. He then enters the teaching activity class room space and assumes a new role, of teacher, using the activity remote control. His badge changes to a red colour to reflect this new role, and the role is listed above the default one (which he is still playing). Various other people arrive at the room, join the activity and assume the student role. Each of their badges turns from blue to white, allowing Steve to check visually that everyone in the room is a student. Steve can now begin to teach. Different objects in the room respond differently to different roles. For example, only Steve would be able to use the blackboard but all the students would be able to view the information on it. Once the lesson is finished Steve no longer needs to be in the teacher role, or the teaching activity, and changes his context by selecting a different badge to wear. Since the teaching activity is quite local (i.e. very few objects in the activity are spread beyond the class room) the actual teacher role has very little use. Therefore Steve "removes" the badge completely and stores it into the database of possible roles carried by the virtual adapter.

14.5.4. Objects

Presentation and sharing

In COLA the amount of sharing (through interface presentation and access control) of any activity object was performed through special objects known as object adapters. These used the state of the client within the objects activity, the clients context, to specify exactly which operations were allowable to the client (with that particular role and involved in a specific activity). The adapter calculates some level based upon the clients context and the state of the owning activity the object is being shared in. This level is then used to dynamically present a subset of the overall interface which accurately reflects at that instant how the client may use the object and what operations are available.

For lightweight activities in a virtual world, the same process still occurs at the underlying platform level, but now the adapter can present the shared object interface based not only in terms of the client and activity status but also *based upon the objects actual spatial coordinates* i.e. how aware the objects are of each other. The detection and subsequent presentation of a client context to the object is performed through the collision of the activity spaces around the objects and the spatial awareness level is obtained through the collision of the two objects focus and nimbus. In essence, this extension of the adapters means the *subset of usable operations by object X of an interface belonging to object Y, is a function of how spatially aware Y is of X* ***and*** *the context X has within the activity of Y* .

Example

Consider two activity members, a lecturer Steve and a student Tom, entering a lecture theatre, containing objects belonging to a lecture theatre activity. These include a shared white board which is positioned on the far wall. The white board has three reactive spaces: nimbus, focus and activity. Each of these spaces surrounds the entire lecture theatre. Steve and Tom exert smaller surrounding activity spaces and a focus space (figure 14.3).

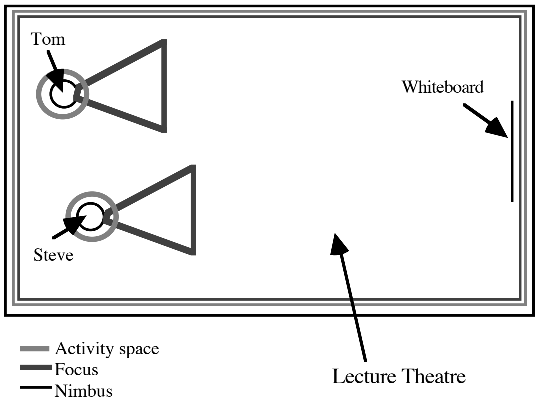


Figure 14.3: Plan view of reactive spaces

As soon as they enter the theatre a number of spaces collide and react:

- Tom and Steves focuses collide with the whiteboard nimbus. The whiteboard then calculates an **awareness level** of Tom and Steve, based upon how "large" a collision it is and Tom and Steves spatial coordinates relative to itself.
- Tom and Steves activity space collides with the whiteboards activity space. The whiteboard "reads" the information stored in Tom and Steves space to get their **activity contexts**.

This information is then used by the whiteboards virtual adapter to present a subset of all the operations to Steve and Tom. In this case both Steve and Tom can see that the whiteboard has information on it but not what that information is. If we had changed this example so that the whiteboard was in a private meeting and Tom was not a member of the activity then the whiteboard may refuse to display any information to Tom because of Toms activity context.

As Steve and Tom approach the board the level of spatial awareness between them and the board increases to the point where more operations are available to Steve and Tom, who can now see all of the information on the board. When Steve and Tom reach the board itself the board uses their activity contexts to decide what operations are allowed (i.e. who can write on the board). Since Tom is only a *student* no operations are provided, he can only see what is written on the board. Steve, being a *lecturer*, is presented with a full interface to the board and can write on it (figure 14.4).

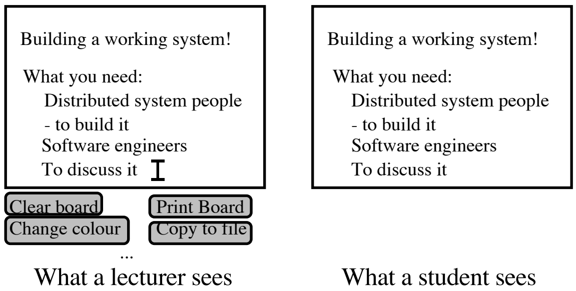


Figure 14.4: Different views based upon activity context

14.5.5. Special spaces: access and navigation

Navigating through various activities would not be much of a problem if an "only one activity per room" regime were implemented: a simple hierarchy displayed on the activity remote control would be sufficient. Unfortunately, it is difficult to see how users can navigate through the activity hierarchy in two dimensions, given that the activity is actually spread across three (or more) dimensions and interleaved with other activities. Therefore we need some other method of helping users navigate through activities. Additionally, the creation of objects and new activities in the COLA platform did not rely on any spatial metrics (since none could be applied). Within a virtual environment the same creation **must** be spatially aware to enable users to apply spatial positions to new objects and activities entering the existing space for the first time.

A virtual world frees the user from the constraints of *having* to use the remote control activity browser in the normal, 2D, way. We make use of this fact by adding special options to the control which activate various gateways. These gateways will take the user to one of a number of special, separate spaces:

- Control room

  This space contains a spatial overview of the state of an activity. The actual physical positions and shapes of the activity objects (members, resources, role badge colours) are displayed. This is done through a combination of lists of objects and their virtual representations and a miniature “god-like” overview of the spaces they can be found in. This reflection of the activity space also obeys the same laws that the original does. Therefore, by manipulation of visors a user may be able to selectively see only the active members in the representation etc. The user is free to examine the state of the objects by moving around the 3D replica and “pointing” at them (figure 14.5)

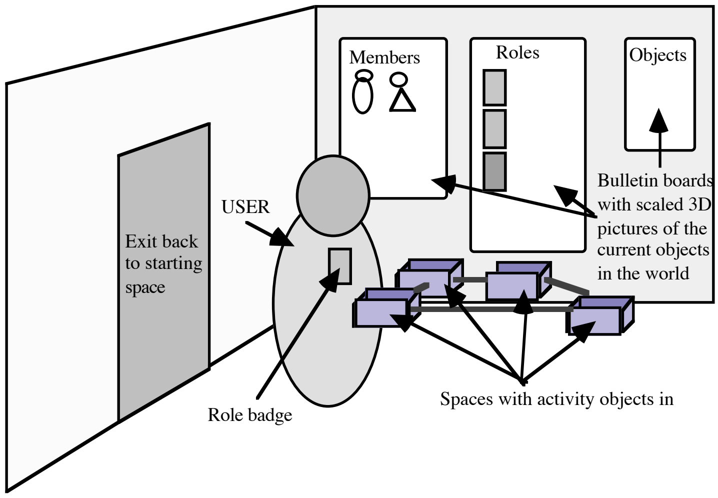


Figure 14.5: A examining the current state of an activity

These control rooms are not necessarily restricted to one user, or even to members of the activity. Activity members may possibly enter these places when they wish to discuss something about the activity in a more visual manner. Further, the remote control can generate gateways to special private “creation” control rooms where users can create new activities and objects **on-line** before starting them.

- Activity information space.

  This space represents a spatial hierarchy of all the activities in the system and the user can traverse and examine each node (which represents an activity control room). Each of these nodes contains the state of the activity which is available for browsing (in the same manner as an activity remote control displays information). Once at a node a user can perform high level activity operations, such as joining or entering a gateway to the control room for the activity at the node. Because the nodes of the space are accurate reflections of the activity control rooms, a control room space which only allows members will also disallow the user in the information space to visit the node. These access rights can be specified on the boundaries of any space using container spaces (explained in section 14.6.3.2). In this manner, any restrictions which exist in the “real” space (where the activities take place) are maintained in this miniature version.

When the user interacts with these spaces by entering a gateway, other objects in the original space would either only see an unresponsive user object “looking” at a small interface tool in their hands or the user would disappear completely. This would depend on the specification of the gateway. In the case of the gateways above, a representation of the user remains. Exiting the information and control spaces would causes the user to be "put back" into the object through which they where manipulating the browser (i.e. their normal representation in the virtual environment).

14.6. COLA support services in a virtual environment

The services that are required to support the COLA model in a virtual environment can be divided into two, depending on what aspect they support:

- Original COLA lightweight activity support services and managers
  - Activity manager, activity and adapter store
  - Event manager, event store
  - Object Adapters, object factory, object store
  - Shared object trader, binder
- General services for a virtual environment
  - Application vendor
  - Virtual adapter server
  - Space server: reactive, container and node spaces

We concentrate on the general services. Readers are referred to (Trevor et al. 1993) for an description of the basic services required by the COLA model and their implementation in the COLA platform.

14.6.1. Application vendor

In order to actually start an application within the virtual world, or to get a remote control object to manipulate and browse an ongoing one, an object interacts with an application vendor. These may be "machines" which stand in special rooms in a virtual environment, or available through the virtual adapters with a vendor remote control. The applications in the environment use the vendor in a similar manner to the ANSA "trader" service (ANSA 1989), by registering themselves, their location and an application remote control (if appropriate). The vendor allows objects to browse its contents which can then join, locate and even start new applications using vendor. Starting or joining an application may result in an application remote control being given to the user by the vendor or the application.

14.6.2. Virtual adapter server

This service provides generic virtual adapter objects for objects entering the virtual environment. These are either people actually entering the virtual space for the first time, or objects which have been newly created. The virtual adapter objects themselves provide the mechanisms through which the user or object employing the suit can change and interact with the many reactive managed spaces in the virtual environment (figure 14.6). Operations include the:

- storage and retrieval of portable objects (badges, visors, spaces etc.) that can be "connected" to the underlying object through the suit.
- presentation of the portable object interface operations
- interaction with the environment services, such as container space managers

These operations can be made externally visible to users through the use of one or more control pads which they always carry around with them.

All virtual adapters should present the same set of operations to the object which they surround and the objects which they connect. This enables users to move through, and interact with, a variety of potentially very different spaces by using different versions of virtual adapters. As long as the core interface presented by the adapter to the underlying object does not change then the object should be able to interact in any environment.

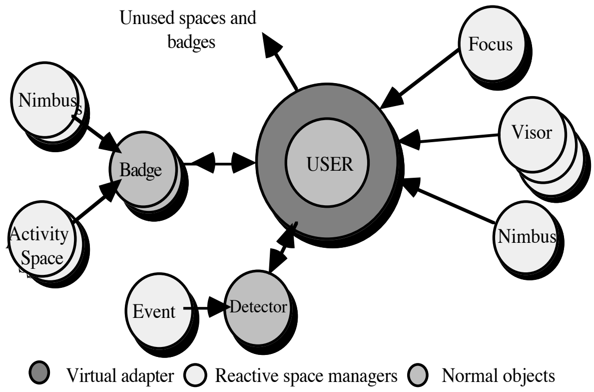


Figure 14.6: A working Virtual adapter

To illustrate an adapter consider a user being initialised in a virtual environment. First of all the user "jacking-in" to the environment asks the virtual adapter server for an adapter. The server generates an adapter which has an activity remote control stored in its internal database of objects. The user presses some external physical button which signals the adapter. This causes the adapters interface to appear in the virtual environment. The user then examines the contents of the virtual adapters database of "pockets" and uses the activity remote control[1]. The adapter removes the control from the inactive database and places the object in the users virtual hand. In functional terms, this may translate an object being brought from backing store and activated in local memory. Any current remote control would be put back into the database. The user then looks at his rendered hand and sees a remote control in it with a flat representation of the current activity hierarchy. He selects an activity to

1 We assume for the sake of the example that the virtual adapter has been "preloaded" with an application remote control. In practive the only remote will probably be to the application vendor service, which the user can use to obtain the activity remote.

join by pressing a button on the remote[1]. This causes the activity remote control to send a join signal to the relevant activity manager.

The activity manager joins the user and assigns him the default role. The manager creates a new role badge object and returns it to the remote control. The remote control then exports this new object to the virtual adapters database and then asks the adapter to activate it, all using operations from the standard adapter interface.

The role badge uses the virtual adapter to get the identity of the underlying object in this activity and the system addresses of any other role objects which are also currently active and attached to the adapter. Since this group is empty, the role adapter assumes it is the active role and activity and requests a reactive space from the space server (below). An event space is then created around the original object which contains the activity context of the user.

14.6.3. Managed space server

This server will produce space managers of all sizes and shapes requested by the client of the server. Two distinct types of space can be requested and produced: *reactive* and *container*.

Reactive space

Reactive space has been described before and produces some result when it comes into contact with other types of space. Exactly what happens and which other spaces / objects it reacts to, are held as a set of "behaviour" templates for the space. These are passed to the space server along with the creation request.

Container spaces

Container spaces are different to reactive spaces as they provide *managed boundary markers* for objects and spaces within the space. In order to enter a container space a number of requirements must be satisfied by an object wishing to join, held in the container manager as a set of possible object templates. These are defined when the space is created but may be subsequently altered as desired by the container space owners (again, templates can be used to specify groups of objects who can "own" this space). In addition to selecting which objects may enter the space, the manager also contains the addresses of the parent space manager and all container managers that monitor subspaces of itself. This information is necessary to actually control someones movement through connected spaces and sub-spaces.

The concept of managed space, both container and reactive spaces, can be applied recursively. There is no reason why a normal reactive space inside of a large space cannot itself generate a container space, which reflects its own space, for other spaces and objects. Membership of a subspace can be exclusive or inclusive of the parent-space depending on the container space manager. If membership exclusive to the manager then the parent-space of the manager has no knowledge of

[1] This is all in an "ideal" virtual environment. Clearly the actual methods of manipulating and presenting the interfaces and remotes must differ as datagloves and concepts such as "pressing" a button are difficult to model.

the object within the subspace. This membership is inclusive then each container spaces passes the information repeatedly upwards to parent spaces until an exclusive membership space is met. This potential hierarchic and layered approach to spaces can be used structure very large virtual worlds.

We propose a special container space known as a *node-space manager*. These are container space managers which form the roots for any space, having no "parent" manager. The "parents" of a node are the entry points for other **separate** spaces and virtual environments. This relationship between spaces, subspaces and separate spaces is shown in figure 14.8. Only objects may be transferred between node-managers and the space managers which they are joined to (shown as grey arrows). No information about space membership flows between these managers. In this manner any real "root" node of the "universe" does not contain information about every object in the space. This is localised to the subspaces below a node-managed space.

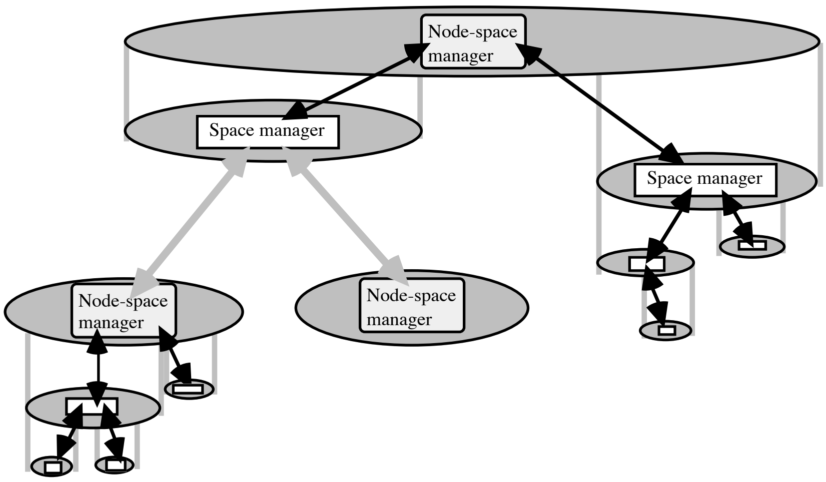


Figure 14.8: A virtual world hierarchy

Using spaces and subspaces to form a large virtual environment captures aspects of work done with both the Domino project (Solman and Moffett 1989) and the X.500 directory service (ISO/CCITT 1988). The Domino distributed systems management project proposes a domain based system for structuring. A domain consists of a set of entities, which are grouped together according to some purpose with a common management policy. Domains may be subdivided into hierarchies with separate policies for each subdomain. These policies are represented by policy objects with general attributes and interfaces which manage various aspects of a domain, including access control. A parallel can clearly be drawn between the domain policy objects and container spaces, each which manage a group of common entities.

In order to navigate through any large hierarchy of spaces, there must be a consistent naming scheme in place. The work on X.500 directory service addresses the problems of naming in order to navigate through an organisational directory. This work seems immediately applicable to many instances of large spatial hierarchies, as it seems sensible to assume that as spaces will often to used to manage real world boundary spaces such as rooms and buildings (and even organisations).

Example of reactive and container space interaction

Consider the following scenario where Steve and John are about to enter a meeting room. Steve and John are represented in the virtual world as objects surrounded by virtual adapters. The meeting room is realised as an empty container space, modelled by a container space manager. Steve tries to enter the meeting room controlled by the container manager. To do this, Steve (or rather his virtual adapter object) must first register himself with it.

The container manager checks the object description presented by Steve against the set of allowable object templates. Since the room is for general use this list is empty and Steve is registered as new occupant. The manager then checks to see if Steves object description is of interest to anyone else in the space who wishes to be informed of any changes (system events). As no-one else is in the room, the manager adds Steve the room group membership. After receiving confirmation that he is allowed in the space, Steve creates an empty "interest" subgroup of the members of the room which is initially empty. This group will be used by **other** objects in the room to receive changes in Steves spatial orientation.

John now tries to enter the room and is allowed entry. John is added to the membership of the space (Steve and John) and creates his own subgroup which, like Steves, is empty.

John and Steve wish to interact with each other in the room and wear two visors (one nimbus and one focus) each. The virtual adapters registers these new objects with the rooms space manager. This causes the room membership to be expanded (John, Steve, Johns Nimbus, Johns Focus, Steves Nimbus, Steves Focus) and the creation of several empty subgroups belonging to the new reactive space manager objects. Figure 14.7 shows the relation between these objects and spaces.

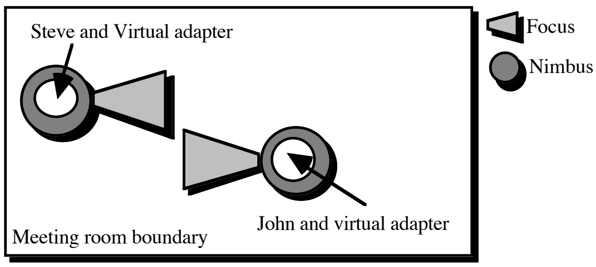


Figure 14.7: Objects and spaces

Each focus and nimbus space then registers with the container space manager the types of managed space and objects they are interested in. The manager checks this interest with all six of the registered object descriptions for possible matches with the interest specified by the focus and nimbus. In this example, both focus spaces are interested in any other reactive spaces in the space and the nimbus spaces are interested in any focus spaces. Each matching interest is added to the "interest" subgroup of the matching object to give:

- Steve and Johns subgroups ()
- Two nimbus subgroups (Steves Focus, Johns Focus)
- Johns focus subgroup (Steves Focus, Steves Nimbus, Johns Nimbus)
- Steves focus subgroup (Johns Focus, Johns Nimbus, Steves Nimbus)

Steve and John move towards each other in order to start a conversation. Each change in spatial coordination of Steve, John and their reactive spaces produces a group event (or message) containing the new coordinates (or boundaries) which is sent to the interest group corresponding to the object, e.g. each change to Steves nimbus is send to Steves Focus and Johns Focus, each change to Steves Focus is sent to both of Johns spaces and Steves nimbus etc..

When these updated coordinates are received by each of the spaces they check to see if a collision has occurred with their own spatial coordinates. If a collision does occur then some awareness calculation is made and a sequence of actions occurs (see section 14.5.4.).

Steve and John finish talking and Steve moves towards the exit to the space. Steves virtual adapter asks (on behalf of Steves object and his spaces) the space manager to leave the managed space and membership of the space group. The manager responds with a list of groups that Steves object and reactive spaces are found and removes them from space membership list (which now contains John, Johns Nimbus and Johns Focus). Steve then leaves each of these groups (or the container space manager does this for him) and destroys his own subgroup. This leaves the groups:

- Johns subgroup (empty)
- Johns nimbus subgroup (Johns Focus)
- Johns focus subgroup (Johns Nimbus)

14.7. Underlying Support

14.7.1. A performance oriented architecture

The physical reality of any large shared virtual environment is that the users will be widely distributed, across buildings, cities, countries and even continents. Ideally, reliable high speed networks would connect everyone together and virtual memory would completely hide this distribution from the services and applications. At this current time however, a more realistic view is a wide variety of local area networks (LANs) connected by wide area networks (WANs) or gateways in order to produce a heterogeneous overall topology. Therefore, any communication between users in such a world must travel over a number of networks with a wide variety of characteristics and quality of service.

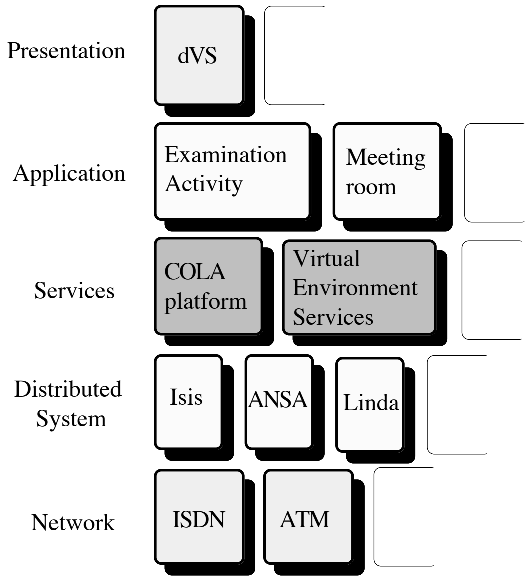


Figure 14.9: Stack architecture

A primary goal of any architecture to support this amount of distribution must be to minimise the amount of world-wide network traffic while still maintaining a coherent and synchronised view of the shared world. We consider how this traffic reduction (or at least restriction to LANs) and overall performance increases can be accomplished at the higher levels of an possible architectural stack (figure 14.9):

- Object replication

  Replication of commonly accessed objects across individual sites and networks enables the virtual environment to dramatically increase the performance for local access of object data. The penalty incurred by managing the replicated states must be carefully weighed against the advantages the approach gives. However, given the inherent latency over large networks this does not necessarily mean that the model should be rigidly synchronised (even if this was possible)

- Duplication of services

  By duplicating key services across local networks it ensures that no object in the environment needs to communicate with a distant service, reducing the traffic travelling on the slower parts of the network. Duplicated services should not need to communicate between themselves very often in order to remain consistent as this would negate the advantages that duplication gives us.

- Model simplification and restriction using spaces

  In an ideal architecture and model, all changes to world objects are sent to all the objects in the same world and these objects then decide if the change has an effect on them. In any virtual environment of any size this quickly becomes unmanageable both in terms of bandwidth and the computational overload that objects will experience. Therefore, this approach to change notification needs to be modified to become more realistic. Indeed, any object in the real world is also bounded in some way, and is normally only aware of near surroundings, such as a meeting room.

- Caching

  The sum of all the information and objects in a large virtual environment is obviously too large to be held in memory all at once. Inevitably, this means some data must be held in the backing store. Unfortunately access to this information is very slow compared to local memory. Therefore, any practical realisation of a virtual world must employ a successful caching strategy to be able decide which part of the virtual world should be held in memory at each network / site / machine and to enable a smooth transition when a user moves from the currently held part of virtual world to another.

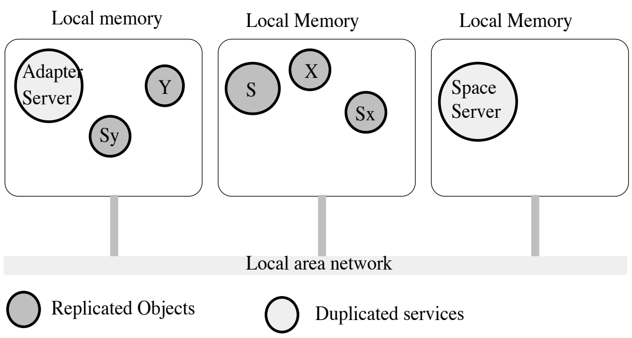


Figure 14.10: System view of object placement

Figure 14.10 shows a local view of the services we have introduced and some active objects. These services (Space server, Virtual Adapter Server) are duplicated on each local area network to localise traffic. The objects (X, Y, Sy, Sx and S) which are frequently referenced and accessed are replicated across the networks (this level of replication and duplication may be extended to each area of local memory if desired). It is the responsibility for the distributed system support to maintain the state of the objects in some way. It is arguable that strict replication is unnecessary between objects because the latency inherent in the networks between the replicas, which may force all changes to become asynchronous in order to keep the system usable. The desire for replication and associated problems are addressed by the dVS architecture (Grimsdale 1991d) where all objects are replicated locally and updates are propagated in such a way as to ensure consistency but not synchronisation.

In order to keep the system level messages (system events) about a new object position and the computational overhead to some manageable level the Comic spatial model employed the concept of aura. When auras collided between objects their was the potential for interaction and only then did the objects involved start calculating awareness levels using their focus and nimbus. This aura, a boundary beyond which actions in the virtual world cannot travel, can be realised as instances of container spaces (section 14.6.3.2). In the spatial model, aura was something every object had. By using container spaces we generalise this concept and allow larger groups of objects to be within a container space (or aura).

The special instances of container spaces, node spaces, reflect the logical boundaries between spaces and do not allow the propagation of system events (only objects). By viewing the logical boundaries between spaces as "physical" joins between the different spaces that make up the large virtual environment we can dictate a caching strategy which relies on users moving between different nodes infrequently. New local world information is loaded from a world cache when a user passes through a node managed space to a new node managed space. All the node managed spaces which are possible to reach from this new space are taken from backing store and cached (if they are not already in the cache). The actual position of this cache depends on the configuration of the network.

14.7.2. Distributed system support

A key distributed requirement for such a literally distributed architecture is a multicast service with a variable quality of service. Sometimes the multicast must be delivered at the same time to all members in the group (e.g. an message that an object has been destroyed need only be sent once but it is important that it arrives so that everyone stops using it at the same time) while other multicasts require a very low grade service is required, e.g. a object moving through a room may only require new spatial coordinates to be multicast. Since that person will probably move again and precise positioning is probably not critical, dropped position packets or latent delivery problems can be ignored.

An obvious candidate for a distributed toolkit to support this is ISIS (Birman and Marzullo 1989, Birman et al. 1989) and indeed the architecture was designed with ISIS in mind. ISIS is a distributed programming system which provides fault tolerant process groups and ordered multicast. Groups are useful for structuring distributed programs and coordinating their activities. Multicast can be used to communicate reliably with all processes in a particular group. Members of an ISIS process group know exactly who the other members are at any given time, so that it is easy to write algorithms that partition work between group members (e.g. for parallelism for load balancing) or so have exactly one group member perform some task while the others stand by as backups and are available for other tasks. Isis also provides a number of potentially very useful mechanisms:

- Automatic failure detection

  If a process running ISIS fails then members of the groups in which the process participated are automatically notified (if they wished to be)
- State transfer mechanism

  When a process joins a group of other processes this mechanism allows the new group members to receive some group-wide common state
- Virtual synchrony

> This unique features of Isis guarantees any multicast the correct delivery order of messages at each member of a process group. This feature, and the ability to relax it in a number of gradual stages[1], is obviously very useful in any distributed environment and when combined with automatic state transfer enables easy data replication.

However, ISIS provided very little support for other aspects of a distributed system, such as type checking, inheritance and an interface description language[2]. This may prove difficult in a virtual environment as a great deal of emphasis is placed upon using objects.

An alternative, the ANSA testbench (ANSA 1989), is based upon the ODP reference model (ISO 1991) and employs a more conventional client-server model of interaction. Indeed, Benford and Fahlen (Benford and Fahlen 1993) proposed a collision detection service using an extension to the ANSA trader. We agree when they argue that the passive trading model is not sufficient for the ad-hoc and opportunistic interaction between objects (especially those representing human beings) and needs to take into account the spatial proximity of the objects. However, we feel that *even* active trading is inappropriate for a number of reasons: the only objects that should be involved in the exchange are those that collide; the actual interfaces which are presented to each of the objects must change dynamically as their awareness of each other alters; and the basic client and server paradigm is simply not be well suited to the demands of a virtual world. The sequence of locate service, make request; await result, use result is almost entirely a one way process with the client dictating all the actions. We argue that interaction between objects in a virtual world is more mutual in nature, and the properties of the interaction are often far more complex.

Another higher level distributed toolkit worthy of mention is Linda (Gelernter 1985). Linda uses a tuple space based methodology for “message passing”. Objects place tuples into some shared space and other objects can remove or copy the tuples out at a later time. While a number of problems exist with this approach, it is interesting to note that the concept of a tuple space could be mapped onto real space coordinates, or even managed spaces and back again. Further work is needed to determine how useful this conceptual overlap might prove to be.

14.8. dVS — a Distributed Virtual environment System

dVS is a system architecture which supports a highly parallel model of computation, designed to facilitate the building of advanced user interfaces (Grimsdale 1991a). The development of a parallel model for virtual environment simulation is needed in order to achieve high levels of performance. The model is based on four

1 Giving a selectable quality of service

2 This lack of suitable object-oriented support may soon be resolved with the introduction of “Reliable Distributed Objects”, an extension of Isis to be released in September. "Reliable Distributed Objects" has interfaces for C++, Objective-C, and Smalltalk and will integrate with regular C Isis programs and is based upon the CORBA IDL as its interface generation language.

different layers of parallelism, the highest of which is *environment*. This enables independent simulation of different environments and are subdivided into autonomous *entities*. These are typically high level 3D objects which encapsulate all the state required to define the visual, audio and behaviour attributes etc.. These attributes are called *elements*, such as position, constraints, force, collision and are processed in parallel. The lowest level of this parallel approach is called *primitive* which controls the rendering of hands, heads etc. In addition to the parallel process model, dVS provides several other key features: Object oriented API; Hardware independence; Scalability; Open system; Support for multiple participants.

A consistent set of services (figure 14.11) are presented through an application programming interface, VL. VL is essentially a distributed database access library enabling distributed processes running in parallel to share global data in a consistent manner. Higher level functions which change the visual and audio properties (among others) are provided by a toolkit called VCTools. Both VL and VCTools are written in ‘C’ enabling the application code to use high-level language functions and operating system calls as well as those to the virtual environment.

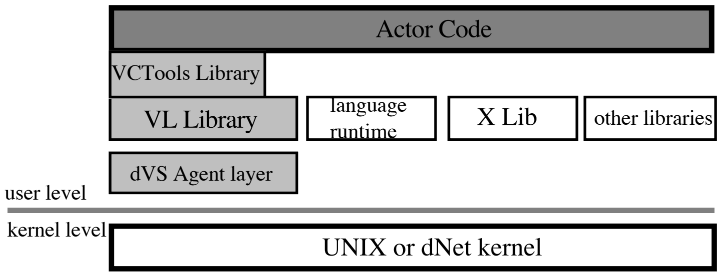


Figure 14.11: dVS system services

14.8.1. dVS model

The dVS model (Grimsdale 1991b) is composed of discrete processes called *actors*. These collectively control the progress of the environment simulation. Actors share global data which is maintained both locally and communally. Each actor maintains a local copy of the global data, but the global data is invisibly shared among all actors by dVS itself.

The fundamental objects containing global data are abstract data types called *elements*. An actual instance of an element is called a *Parcel*. An actor which references a parcel is said to *hold that parcel*. All references are local (so each actor referencing holds a “local” instance of an element). Each actor executes asynchronously to every other, so different actors holding the same parcel may hold a different state. When an actor wishes to change a parcels state it makes an explicit update request which is propagated by dVS to all the actors holding the parcel.

As dVS executes on loosely coupled networks (with non-deterministic transport mechanisms) it is impossible to predict how long these updates may take to filter to all actors holding a parcel. It is undesirable to synchronise all the parcels state at a specific time since performance and response time under heavy load could be very slow. Therefore each actor proceeds in its own time domain, so actors with the same parcel may hold different state. A special actor called a *Director* acts as a parcel cache and mediates parcel updates to each actor holding a the parcel (an instance of the global data element). Directors provide a medium through which different actors remain approximately synchronised.

Global time in a distributed system is notoriously difficult to maintain and initialise. dVS uses the concept of *virtual time* which is kept by each actor in an environment. An actors virtual time is initialised by the environment director which sends out periodic adjustments to each actor to keep the actors in close, but not exact, synchronisation with its own time. These adjustments to a specific actor are calculated based upon the network delay between the director and the actor and the value of an actors virtual time which is passed with each parcel update.

Finally, an *environment* consists of a collection of actor processes and parcels, managed by at least one director. Currently the model has no access control on parcel state, although restricted holding of parcels is proposed through encrypted capabilities to allow selective access to certain elements and parcels.

14.8.2. Using dVS to support the visualisation of lightweight activities

Very few virtual programming environments or operating systems exist. Examples of these include dVS and DIVE (Fahlen et al. 1993). dVS seems an ideal choice to support the visualisation of lightweight activities over these other approaches because it enables existing applications and functions to be incorporated through a common language (‘C’) and libraries (VCTools etc.).

All of the underlying problems of building, representing and implementing 3D objects in the virtual world are hidden from the programmer using the dVL interface (Grimsdale 1991b). This provides a large number of function calls, including registering with an environment, creation of parcels, collision detection, creation of actors/objects, positioning etc.. Therefore, the actual representation of the ideas described in this report are well supported since this is *precisely* what dVS aims to do.

Virtual adapters, visors, role badges and remote controls can be built as one or more actors, some of which provide the actual functionality of the object and other which manage their own image and interfaces in the environment. For example, the presentation of a suitable virtual adapter interface is a simple extension of the dVS menu and pointer example given in (Grimsdale 1991b). Several different actors are combined to allow a menu (a simplification of the virtual adapter interface) actor to interact with associated pointer and visual actors, which control and visual the pointer to the menu respectively.

Again, the actual representation of managed spaces is also relatively easy realisable. Using a general framework for an actor with an *element* of type “space”, which contains the shape that the space takes up, we can construct and visualise any shaped reactive space desirable. However, using dVS to capture the actual *functionality* of reactive spaces, as well as the representation, presents the first of two (apparently) large problems.

In dVS, the presentation of a virtual environment comes from a single version of the whole global environment state maintained among the system agents. Each actor sees a specific subset or reflection of that state by registering interests in the various elements of data which make up the environment state. These elements are replicated as parcels at each actor in order to increase the system performance. Basically there is *only one method of viewing the world* through replicated world information. Unfortunately, this approach appears to give *no support for the presentation and sharing of objects* ***differently*** *to different people*, e.g. the reaction of any reactive managed space with an object may be visual (it flashes etc.). This change in the objects visible state should be localised to the object projecting the managed space and no-one else. The same underlying system object is being presented in a different manner at the same time.

One method to overcome this apparent shortcoming is to replicate **every** reactive space element (which models the shape of the physical space) and **every** object visual element as parcels for each users actor in the space. If a collision occurs between a space managed by a user actor and another space then the visual element of the space/object can be rapidly turned on or off (or change colour). As the parcel state of any element does not have to be propagated to the other parcels, these changes to the local parcel may be deliberately "lost" to other replicas. However, this solution may prove to be too performance limiting.

The second problem with dVS becomes apparent when we consider how dVS objects interact between themselves. dVS relies on replicated state for actors to communicate and interact with the environment. Little support is actually given to aid **any** interaction between the objects themselves (other than through access to a shared state) or the externalisation of an interface. This means flexible interface sharing and dynamic presentation between environment actors is almost impossible just using dVS, since the only means an object has of interacting is through changes to replicated data (parcels). In our opinion, dVS actors and objects are simply **too** autonomous, with little provision for the actual interaction of objects.

14.8.3. Underlying support for lightweight activities

In section 14.7 we outlined a performance oriented architecture to support lightweight activities. The ISIS toolkit seemed a natural choice for supporting such a group oriented architecture and provides many other useful distributed mechanisms but is a general toolkit and supports no specific problem domain. Conversely, dVS is a specific operating system, designed to support the construction and use of a virtual environment across several software layers (figure 14.11) (Grimsdale 1991c).

The specialised support provided by dVS has many advantages over the more general ones. In addition to the easy manner in which objects may be represented in a virtual environment, and the parallel approach of the architecture, dVS provides other useful features and services: an elegant method of keeping distributed time in a closely synchronised manner; collision detection; the ability to use a standard service independent interface to detect collisions, movement etc. All of which are invaluable to a virtual applications developer.

However, there seem to be a number of shortcomings if only dVS is used as an architecture to support virtual environments. The biggest, the lack of any means of inter-object communication and interface presentation, have previously been explained. However, the current dVS model of environments, actors and elements has no notion of *ownership*, or more generally access control, on shared elements (parcels) and actors. Consequently, everyone can hold any parcel without any restriction. In **any** multi-user virtual environment where people are interacting between themselves and objects, these interactions need some degree of control, even at the most primitive level.

In essence, dVS attempts to provide support for virtual spaces at all levels of the system, from the actual presentation of the objects in the world (which it does very well) for a user, to the mechanisms to support replicated data and message passing on networks with low grades of service. The dangers of developing a specialised and all-encompassing approach to support virtual environments are perhaps analogous to the problems evident in an majority of cooperative support systems, such as CHAOS (De Cindio et al. 1986), COSMOS (Araujo et al. 1988), Amigo (Danielson et al. 1986), OTM (Lochovsky et al., 1988) and Rendezvous (Patterson et al.). Far too often these systems are rendered unusable because they rigidly support a specific subset of the problem domain. Consequently, the final systems tend to be characterised as *heavyweight* with unrealistic models of the real world, constraining models of control, lack of group “awareness”, and a limited support for sharing (Trevor et al. 1993). Indeed, (Benford and Fahlen 1993) also describe these work-flow and process oriented systems as too rigid and unnatural for supporting conversation management in spatial settings. However, it appears that the developers of dVS have averted the potential pitfalls of attempting to provide complete support by allowing the user access to the underlying operating system and other C library routines. In our opinion, the ability to use a variety of other libraries and toolkits, such as ISIS, to support virtual environments should not be seen as merely optional, but as *very necessary additions to any virtual environment support system*.

14.9. Further work

Virtual environments have enormous potential and the directions future work can take reflect this, at all levels of an architecture. At the most basic network level there are two crucial problems which manifest themselves in all subsequent layers and are therefore not easily dismissed; *bandwidth* and *latency*. While replication, duplication and bounded spaces all help to lessen the potential network traffic and speed up access, other aspects such as integrating real-time audio and video into a virtual environment will drastically increases the required bandwidth. Arguably more important still, is latency. Actual delays in reflecting real-world movements and actions in the virtual environment may render the interface, and therefore the system, unusable (a conclusion reached by many people about any sort of interface, including (Greenberg et al. 1992)). Further, the problems in synchronising replicated objects and actions where significant latency is present are potentially very worrying. For example, two users are interacting with a replicated object. If the replicas receive the same changes in state at *different* times, the objects state will become inconsistent. While toolkits such as ISIS (through virtual synchrony) can solve many of these problems, the price of such solutions may not be easily dismissed.

The full potential and impact of reactive managed spaces has yet to be explored. The use of spaces as visual identifiers (objects which flash to indicate which activity they are in for example) raises a number of interesting possibilities, such as invisible objects. What are the consequences of an object which can only be detected through the interaction of two reactive spaces? Does such an object "occupy" the standard three spatial dimensions or can it be walked through?

Another attractive possibility for managed spaces are *history spaces*. These can be linked to objects and react with every space which comes in contact with that object. This reaction is recorded in an "interaction" script which the space itself maintains. Other objects can then see the "fingerprints" of the old interactions by projecting a reactive space which "examines" the history script. This would allow users to wear a visor which tells them who actually used the object last, or passed by it. Further, history spaces without an associated object can be viewed as "traces", or "vapour trails". These could be left behind (and exist for some period of time) whenever a person moves through a space. Again, detection of these spaces through visors could enable a user to be "followed", or to retrace their steps (a popular Greek legend involving a minotaur leaps to mind!)

Interestingly, while an obvious use of a container space is as a room, container spaces can also represent the physical constraints of the underlying network. For example, given that an object template is sufficiently detailed, the user could set up a space which restricted the membership to all objects with a local network address. This would effectively create a space which was entirely local and consequently respond very quickly (since there is no synchronisation or communication with physically distant objects). Therefore, container spaces can be used to actually reflect the underlying "physical boundaries" of a system at a very high level, or application generated ones. Further still, the ability for a container spaces to restrict "passage" through, or into it, may be invaluable in supporting the notion of boundaries in the Comic spatial model.

Some other future work highlighted by this report includes:

- Alternatives to ISIS, such as Linda and a group supporting ANSA testbench
- Magic wand feature of navigating through space(s) — section 4.3
- Other virtual adapter attachable environment interaction tools

And last, but by no means least, future work may well include actual prototyping of many of the ideas presented in this chapter.

14.10. Summary and conclusion

This chapter has demonstrated how the main features of a model for lightweight activities, COLA, can be visualised in a virtual environment. This was done using a combination of the spatial metaphor proposed bin Chapter Four, coupled with several other ideas:

- Reactive managed spaces

  Reactive Spaces can be used to control many different types of interaction between virtual objects, e.g. nimbus and focus spaces provide conversation management. On collision with other objects and spaces, reactive spaces perform some set of actions, such as the exchange of interface references.

- Virtual adapters

  Virtual adapters aid objects by supporting the use of general virtual tools and spaces. By interacting with its adapter, an object can change how the virtual environment is perceived and how the environment reacts to the object (or visa-versa). The adapter also interacts with the underlying system on behalf of its owning object.

- Visors

  Visors are used by objects primarily through connection to the object virtual adapter. A combination of visors can be used project and visualise a selection or subset of the objects and spaces in the environment.

- Remote Controls

  In any general computer system, a large number of applications are available at any moment in time. Users should be able to easily move between these different applications as desired. In a spatial world it is sensible for frequently used applications to travel, in some way, with the user. One of these ways is provided by application specific remote controls, which act as remote interfaces to applications, no matter where the user is.

- Gateways

  These are generally applicable to any virtual environment, but in our example, are activated by activity remote controls to allow instantaneous travel to different managed spaces.

These tools and spaces were used to express the main concerns of the lightweight activity model in a visual, and malleable, manner:

- Activities

  Activities are modelled as reactive spaces around activity objects (members of the activity). The externalisation of the current activity state is managed through an activity remote control, which serves as a two dimensional browser linked to some distributed activity manager.

- Roles

  Roles are visualised as coloured badges which people wear to play them. The current role of any object belonging to more than one activity is determined by the top-most badge.

- Events and virtual events

  The notion of COLA events remains the same, with member objects having activity events delivered directly to them. We identify a new type of event, a virtual event, which makes use of the spatial metaphor. These events can actually travel through a virtual environment, present reactive spaces, and interact with their surroundings.

- Object sharing

  The degree of sharing in COLA was determined by the context the client of the object had, within the objects activity and the actual activity state itself. The ability to change the level of sharing based upon an awareness of the objects surroundings (activity and client) was extended to incorporate the notion of spatial awareness. This allows an object to present different subsets of available operations to clients depending on their *activity state* and how *spatially aware* of each other they are.

In addition to these original lightweight features, the activity remote controls provide gateways to special rooms and spaces where the spatial visualisation of a hierarchy of activities, and the presentation of individual activity state, can take place.

The underlying support for this visualisation comes from several services (application vendors, virtual adapter and spaces servers) and at a more fundamental level, a performance oriented architecture. Performance benefits were gained using a variety of techniques: replication of world data; duplication of services; and managed boundary spaces, which restrict bandwidth and provide a sensible cashing mechanism for the entire environment. The model of reactive space collision detection and increased performance requires a variable quality of service multicast from underlying distributed system. This gives adequate support for data replication and rapid information propagation around the system. The group paradigm employed by ISIS was an obvious candidate to provide such a service.

The report then showed how dVS, a specialised operating system for distributed virtual environments, could be used to visualise COLA and outlined a number of possible problem areas, the foremost of which seems to be that dVS objects are too autonomous, with little provision for explicit inter-object communication and information exchange.

In conclusion, we have demonstrated how a lightweight activities may be used to structure and control the natural free-style model of conversations that take place in a virtual environment. This was accomplished using a variety of concepts and tools, which also are applicable to any general virtual environment. While a great deal of further work is required with many of these issues, concepts and tools in this report, the range of possibilities for any form of managed space in virtual environments should not be ignored. Reactive spaces can naturally model interaction between objects in the environment, while container and node spaces lend structure to the world from the most basic level of the network to the visualised rooms that people may inhabit.

References

Acker, S. and Levitt, S., (1987). Designing Videoconference facilities for improved eye contact, *Journal of Broadcasting and Electronic Media,* 31, 2, Spring 1987, pp 181-191.

Ackerman M.S. and Malone T.W., (1990). Answer Garden: A tool for Growing Organisational Memory, in Lochovsky F.H.(ed) *COIS90 Proceeding conference on Office Information Systems, April 25-27, 1990 , Cambridge, Mass.*

Ackroyd, S., Harper, R., Hughes, J.A., Shapiro, D., Soothill, K. (1992), *New Technology and Practical Police Work,* Milton Keynes, Open University Press.

Ahlström B, Marmolin H. & Marmolin T., (1993). Ongoing Evaluation Studies of Collaborative Work Within the Swedish MultiG Research Program, *Proceedings of INTERCHI 1993, Amsterdam, The Netherlands, 24 - 29 April, 1993.*

Ahuja, S. R., Ensor, J. R., Horn, D. N., (1988) The Rapport Multimedia Conferencing system, in *Proceedings of the Conference on Office Information Systems* (COIS88), Allen R. B. (ed.), March 23-25, Palo Alto, Ca., 1988.

Alexander (1989),A City is Not a Tree, in *Zone 1/2*, Michael Feher and Sanford Kwinter (eds), Urzoen Inc. pp128-157.

Anderson, R. and Sharrock , W. (1993) Can organisations afford knowledge? *Computer supported cooperative work,* 1, 143-162.

Anderson, R.J., Hughes, J.A. and Sharrock, W. (1989),*Working for Profit: The Social Organisation of Calculation in an Entrepreneurial Firm,* Aldershot, Avebury.

ANSA (1989), ANSA: An Engineers Introduction to the Architecture, Projects Management Limited, Poseidon House, Castle Park, Cambridge, CB3 0RD UK, November.

Araujo, R.B., Coulouris, F., Onions, J.P., Smith, H.T. (1988), *The Architecture of the Prototype COSMOS Messaging System*., Elsevier Science Publishers B.V., North-Holland.

Bannon, L., Schmidt, K. (1992): Taking CSCW seriously, *CSCW: An international Journal,* vol. 1, no. 1, Oct 1992, Kluwer.

Barghouti, N.S., Kaiser, G.E., (1991) Concurrency Control in Advanced Database Applications, , *ACM Computing Surveys* 23(3), pp. 269-317, September 1991.

Benedikt, M., (1991), Cyberspace: Some Proposals, in *Cyberspace: First Steps*, Michael Benedikt (ed), MIT Press, 1991, pp 119-225, 273-302.

Benford, Prinz, Mariani, Rodden, Navarro, Bignoli, Brown, Naslund, MOCCA–A Distributed Environment for Collaboration, in *Proceedings of Telepresence93*, Lille, France, March 1993

Benford, S., (1993), *Populated Information Terrains,* COMIC Working Paper, COMIC-NOTT-4-4, 1993.

Benford, S., Fahlén, L. (1993), A spatial model of cooperation in large virtual environments, *Proceedings of ECSCW93*, Milan, September 1993. Kluwer.

Benford, S.B., Bullock, A., Cook, N., Harvey, P., Ingram, R., Ki Lee, O. (1993), From Rooms to Cyberspace: Models of Interactions in Large Virtual Computer Spaces. *Interacting with Computers* , vol.5, no. 2, 1993, pp.217-237.

Bentley R., Rodden T., Sawyer P., Sommerville T. (1992) A prototyping environoment for Dynamic Data Visualisation, *Proceedings of the 5th IFIP TC2\WG2.7 Working Conference on Engineering for Human-Computer Interaction*, 10-14 August, C. Unger and J. A. Larson (eds), Elsevier Science, 1992.

Bentley R., Rodden T., Sommerville I,Sawyer P (1992) An architecture to support tailorable multi-user displays, *Proceeding of CSCW92*, Toronto, November 1992, ACM press.

Berlin L. & Jeffries R. (1992). Consultants and Apprentices: Observations about Learning and Collaborative Problem Solving. *Proceedings of CSCW92*, Toronto, November 1992, pp 130-137.

Birman, K. (1985), Replication and Fault Tolerance in the ISIS System, in *Proc. .12th ACM Symposium on Operating Systems,* December 1985, pp. 79-86.

Birman, K., and Cooper, R., (1991), The ISIS Project: Real Experience with a Fault Tolerant Programing System, in *Proc. Workshop on Fault Tolerant Distributed Systems*, New York, 1991, ACM Press.

Birman, K., Marzullo, K. (1989): ISIS and the META project. *Sun Technology ,* no. Summer, 1989, pp.90-104.

Birman, K.A., T.A., J., Kane, K., Schmuk, F. (1989): *ISIS - A Distributed Programming Users Guide and Reference Manual*., Department of Computer Science. Cornell University.

Bly, S. A. and Minneman, S. L. (1990), Commune: A shared drawing surface. *Proceedings of the Conference on Office Information Systems*. pp184-192.

Borning, A. and Travers, M., (1991),Two Approaches To Casual Interaction Over Computer and Video Networks, in *Proceedings of CHI91*, New Orleans, 28 April-2 May 1991, ACM, New York

Bowers J., (1993) Modeling awareness and interaction in virual spaces. In the *proceedings of the 6th MultiG workshop*, Stokholm-Kista, May 17, 1993.

Bowers, J., (1993), *Boundaries, Awareness and Interaction in Virtual Spaces: The COMIC spatial model and representing organisational context,* COMIC Working Paper, COMIC-MAN-1-3, 1993.

Brothers, L., Sembugamoorthy, V. and Muller, M., (1990), ICICLE: groupware for code inspection. *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW 1990).*

Brown, C. and Fahlén, L., (1992), The Use of a 3D Aura Metaphor for Computer Based Conferencing and Teleworking, *Proceedings of 4th Multi-G Workshop,* May 12th 1992, Stockholm-Kista, Björn Pehrson (ed), pp. 69-74

Bush V. (1945). As we may think. *Atlantic Monthly,* 1945:176:101.108

Button, G. (ed), *Ethnomethodology and the Human Sciences,* Cambridge, Cambridge University Press.

Callon, M., Law, J. and Rip, A., (eds), *Mapping the Dynamics of Science and Technology*, Macmillan press, ISBN 0-333-37223-9.

Card, S. and Henderson, A., (1987), A Multiple Virtual-Workspace Interface to Support User Task Switching, in *Proceedings of CHI+GI87, Human Factors in Computing Systems IV*, Toronto, Canada, 5-9 April 1987, North Holland.

Carlsson, C and Fahlén L.E.(1993a), Integrated CSCW Tools within a Shared 3-D Virtual Environment, Formal Demonstration INTERCHI 1993, Amsterdam, April 1993.

Carlsson, C. and Hagsand, O. The MultiG Distributed Interactive Virtual Environment, *In Proc. 5th MultiG Workshop*, Stockholm, December 18, 1992.

Carlsson, C., and Hagsand, O.(1993b), DIVE — a Multi-User Virtual Reality System *Proceedings of VRAIS 1993*, Seattle, Washington, September 1993.

Carroll J, Mack L. & Kellog W. (1988). Interface Metaphors and User Interface Design. In *Handbook of Human -Computer Interaction*. M. Helander (ed), Elsevier Science Publisher B.B (North-Holland), 1988.

Chalmers, M. and Chitson, P., (1992), Bead: Explorations in Information Visualisation, in *Proc. SIGIR92*, published as a special issue of SIGIR forum, ACM Press, pp. 330-337, June 1992.

Chris Condon, *The Computer Wont Let Me*, BICC Systems Development Centre, presented at CSCW: Cooperation or Conflict? DTI CSCW Special Interest Group, London, 23rd October 1991.

Codella, C., Jalili, R., Koved, L., Lewis, J.B., Ling, D.T., Lipscomb, J.S., Rabenhorst, F.A., Wang, C.P., Norton, A., Sweeney, P., Turk, G., (1992), Interactive Simulation in a Multi-Person Virtual World, In *Proc. CHI92*, ACM, 1992.

Comic (1993): Requirements and Metaphores of Interaction in Large Virtual Spaces, Esprit Based Research Project 6225, August 10, 1993.

Condon C., (1990), Networked Cooperative Work, *Proceeding Telematic Workshop,* BIBA, University of Bremen, December 1990.

Cook, S., Birch, G., Murphy, A. and Woolsey, J., Modelling Groupware in the Electronic Office, *International Journal of Man Machine Studies*, 34, pp 369-393, Academic Press.

Cool, C., Fish, R. S., Kraut, R. E., & Lowery, C. M. (1992). Iterative design of video communication systems. In J. Turner & R. Kraut (Ed.), *CSCW 1992*, (pp. 25-32). Toronto: ACM Press.

CORBA The Common Object Request Broker: Architecture and Specification, OMG Document Number 1991.12.1, Revision 1.1, OMG, Draft 0, December 1991.

Crowley, T., Milazzo, P., Baker, E., Forsdick H., Tomlinson R., (1990) MMConf: An infrastructure for building shared multimedia applications, in *Proceedings of CSCW 1990*, October 7-10, Los Angeles, Ca., ACM, 1990, pp 329-342.

Curtis B, Krasner H & Iscoe N.(1988). A field study of the software design process for large systems. *Communications of the ACM,* Vol. 31, no. 11, November 1988.

Danielson, T., Pankoke-Babatz, U. (1988): The Amigo Activity Model., in R.Speth (ed.):*Research into Networks and Distributed Applications*. Elsevier Science Publishers B.V., North Holland, pp.227-241.

Danielson, T., Pankoke-Babatz, W., Patel, A., Pays, P., Smaaland, K., Speth, R. (1986): The Amigo Project - Advanced Group Communication Model for Computer-Based Communications Environment. in (eds.): *Proceedings of the conference for CSCW86,* Austin, Texas, 1986, pp.115-142.

De Cindio, F., De Michelis, G., Simone, C., Vassallo, R., Zanaboni, A.M. (1986): CHAOS as Coordination Technology. in (eds.): *Proceedings of the conference on CSCW86,* Austin, Texas. Dec.3-5, 1986, pp.325-342.

Dollimore, J., Wilbur, S. (1991): Experiences in Building a Configurable CSCW System., in J.M.Bowers and S.D.Benford (eds.):*Studies in Computer Support Cooperative Work*. Elsevier Science Publishers B.V, North Holland, pp.173-181.

Dourish, P. and Bly, S., (1992), Portholes: Supporting Awareness in a Distributed Work Group, *Proc. CHI92,* Monterey, USA, May 3-7 1992, pp. 541-547, ACM Press.

Ellis, C., Gibbs, S. J., Rein, G. (1988) Design and use of a group editor, Technical report STP-263-88, MCC, September 1988.

Ellis, C.A., Gibbs, S.J., and Rein, G.L., Groupware: Some Issues and Experiences, *Communications of the ACM*, Vol. 34, No 1, January 1991, 39-58.

Elmer-Dewitt, P. (1993). Whos reading your screen? *Time International*, 141(3) 30.

Engelbart D. (1990). *Knowledge-domain Interoperability and an Open Hyper-document System*. Proceedings of CSCW90, Los Angeles, October 1990, pp 1-14.

Eriksson H., Frecon E., Carlsson C., (1992), Audio and Video Communication in Distributed Virtual Environments, In *Proc. 5th MultiG Workshop,* Stockholm, Dec. 1992.

Fahlén, L. E. and Brown, C.G., (1992) The Use of a 3D Aura Metaphor for Compter Based Conferencing and Teleworking, In *Proc. 4th Multi-G Workshop*, Stockholm-Kista, May 1992, pp 69-74.

Fahlén, L., (1991),The Multi-G TelePresence System, in *Proceedings of the 3rd Multi-G Workshop,* Yngve Sundblad (ed), Stockholm, December 17th 1991, pp 33-57, Norstedts Tryckeri AB, Stockholm.

Fahlén, L., Brown, C., Ståhl, O., Carlsson, C., (1993),A Space Based Model For User Interaction in Shared Synthetic Environments, in *proceedings of Inter CHI93,* Amsterdam, April 1993, ACM Press.

Fish R.S., Kraut R.E., Leland M.D., Cohen M. (1988) Quilt: A colloborative tool for cooperative writing, in Allen R.B.(ed) *COIS88 Proceedings of conference on Office Information Systems,* March23-25,1988, Palo Alto, California.

Fish, R. S., Kraut, R. E., & Chalfonte, B. L. (1990). The VideoWindow system in informal communications. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW 1990),* (pp. 1-11). Los Angeles, California: ACM Press.

Foster, G., and Stefik, M., Cognoter, Theory and Practice of a Colab-orative Tool, in *Proc. CSCW86*, ACM Press, 1986, pp 7-15.

Foucault, M. (1977) *Discipline and punish*. Harmondsworth: Penguin.

Furnas, Generalised Fisheye Views, in *Proc. SIGCHI86*, ACM Press, pp 16-23, 1986.

Garfinkel, H. (1967), *Studies in Ethnomethodology,* Englewood Cliffs, Prentice Hall.

Gaver W., The Affordances of Media Spaces for Collaboration, In *Proc. CSCW92,* Toronto, November 1992, ACM Press.

Gaver, W., Moran, T., MacLean, A., Lövstrand, L., Dourish, P., Carter, K., Buxton, W., (1991),Working together in Media Space: CSCW Research at EuroPARC, in *CSCW, The Multimedia and Networking Paradigm, Proceedings of the Omnicom Seminar,* London, July 1991

Gaver, W., Moran, T., MacLean, A.,Lovstrand, L., Dourish,P., Carter, K. and Buxton W., (1992), Realising a Video Environment: EuroPARCs RAVE System, In *Proc. CHI 1992 Human Factors in Computing Systems,* Monterey, Ca., USA, May 1992.

Gelernter, D. (1985): Generative Communication in Linda. *ACM Transactions on Programming Languages and Systems* , vol.7, no. 1 (January),1985, pp.80-112.

Gentner, D. R., & Grudin, J. (1990), Why good engineers (sometimes) create bad interfaces. *Proceedings CHI 1990,* Seattle. 277-282

Gibbs S.J. (1989), LIZA: An extensible groupware toolkit. in*Proceedings of the SIGCHI Human Factors in Computing Systems,* Austin, Texas, ACM Press, pp. 29-35.

Giddens, A. (1984) *The constitution of society*. Cambridge: Polity.

Goffman, E. (1961) *Asylums*. Harmondsworth: Penguin.

Goffman, E. (1972) *Interaction ritual*. London: Allen Lane.

Goldstein, I. (1981), Writing with a computer. In *Proceedings of the Third Annual Conference of the Cognitive Science Society*, Berkeley:

Greenberg, S., Bohnet, R., (1991), GroupSketch: A multi-user sketchpad for geographically-distributed small groups, in *Proceedings of Graphics Interface*, June 5-7, Calgary, Alberta.

Greenberg, S., Roseman, M., Webster, D., Bohnet, R. (1992), Human and Technical factors of distributed group drawing tools. *Interacting with Computers* , vol.4, no. 3,1992, pp.364-395.

Grief, I. and Sarin, S., (1987), Data sharing in group work, *ACM Transactions on Office Information Systems, 5* (2): 187-211.

Grimsdale, C. (1991a), dVS - Distributed Virtual Environment System. in (eds.): *Proceedings Computer Graphics,* London, 1991.

Grimsdale, C. (1991b), dVS Overview, 1991, DIVISION Ltd.

Grimsdale, C. (1991c), SuperVision: A parallel Architecture for Virtual Reality, 1991, DIVISION Ltd., Quarry Road, BS17 6AX, U.K.

Grimsdale, C. (1991d): Virtual Reality - key technologies, problems and emerging solutions, 1991, DIVISION Ltd.

Gurwirsch, A. (1979), *Human Encounters in the Social World,* Pittsburgh, Duquesne University Press.

Gust, P., (1988) Shared X: X in a distributed group work environment, *Proceedings of the 2nd Annual X conference*, MIT, Boston, January 1988.

Haake, J. M. and B. Wilson. (1992), Supporting Collaborative Writing of Hyperdocuments in SEPIA . *Proceedings of ACM CSCW92 Conference on Computer-Supported Cooperative Work*.

Hägerstrand, T. (1975) Space, time and human conditions. In A. Karlqvist (ed.) *Dynamic allocation of urban space*. Farnborough: Saxon House.

Hahn U., Jarke M., Kreplin K.et al. (1991) CoAuthor: A hypermedia Group Authoring Environment in J.M. Bowers and S.D. Benford (eds): *Studies in Computer Supported Cooperative Work. Theory,* Practice and Design, North-Holland, Amsterdam, 1991.

Halasz, F., & Moran, T. P. (1982). Analogy considered harmful. In *Human Factors in Computer Systems Conference, 82,* (pp. 383-6). National Bureau of Standards, Gaithersburg, Maryland.

Heath C. & Luff P.: (1991) Collaborative Activity and Technological Design: Task coordination in London Underground Control Rooms, In *ECSCW ´91*. Proceedings of the second European conference on Computer Supported Cooperative Work. Dordrech: Kluwer.

Heath, C. and Luff, P. (1991) Collaborative Activity and Technological Design: Task Coordination in London Underground Control Rooms. In L. Bannon, M. Robinson and K. Schmidt (eds),*Proceedings of ECSCW91*, Sept 25-27, Dordrecht: Kluwer.

Henderson, A., Card, S., Rooms: The Use of Multiple Virtual Workspaces To Reduce Space Contention in a Window-Based Graphical User Interface, *ACM Transactions on Graphics,* 5, 3, pp. 211-243

Hughes, J. and King, V., *Paperwork*, COMIC Working Paper, COMIC-LANCS-4-1, 1992.

Hughes, J., Randall, D., Shapiro, D., (1992) Faltering from Ethnography to Design, *Proc. CSCW '92,* Toronto, Nov. 1992, ACM Press.

Ingram, R., (1992)*Signposting In Virtual Worlds - Some Notes*, 1992.

Ishii, H. (1990). TeamWorkStation: Towards a seamless shared space. *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW 1990)*. pp 13-26.

Ishii, H., & Arita, K. (1991). ClearFace: Translucent multiuser interface for TeamWorkstation (Research report No. NTT Human Interface Laboratories.

Ishii, H., and Kobayashi, M., (1992), ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact, In *Proc. CHI92*, ACM, New York, 1992, pp. 525-532.

Ishii, H., Ohkubo, M. (1991): Message Driven Groupware Design Based on an Office Procedure Model, OM-1. *Journal of Information Processing* , vol.14, no. 2,1991, pp.184-191.

ISO and CCITT, *Information Processing Systems - Open Systems Interconnection - The Directory*, ISO 9594-108, CCITT X.500-X,521, 1988.

ISO/IEC, *Basic Reference Model of Open Distributed Processing, Working Document on topic 9.1 - ODP Trader*, December 1991, Available through national standards bodies.

ISO/IEC, *Basic Reference Model of Open Distributed Processing, Working Document RM-ODP - Part 1: Overview, December 1991*, Available through national standards bodies.

Johansen, R. (1988), *Groupware: Computer Support for Business Teams,* New York, The Free Press.

Kaplan, S.M.(1993), Introduction to the special section on CSCW92 Developers Workshop, *ACM SIGOIS Bulletin*, Vol. 6 No. 4, Autumn 1993, ACM Press.

Kaplan, S.M., Tolone, W.J., Bogia, D.P., Bignoli, C. (1992): Flexible, Active Support for Collaborative Work with Conversation Builder. in: *Proceedings on the conference for CSCW92,* Toronto, Canada, 1992, pp.378-385.

Kedziersky B (1988). Communication and management support in system development environments. In I Greif (ed.) *Computer Supported Cooperative Work*, M Kaufman, San Mateo, 1988, pp 335-367, 1988

Knister, M.J., Prakash, A. (1990), DistEdit: A distributed toolkit for supporting multiple group editors. in: *Proceedings of the Conference on Computer Supported Cooperative Work October 7-10,* Los Angeles, California, 1990, ACM Press.

Korfhage, R.R., Browser, (1986), A Concept for Visual navigation of a Database, Visual Languages, *IEEE Computer Society,* June, 1986, pp 143-8

Kraut R., Galegher J., Egido C. (1986), Relationships and tasks in scientific research collaboration. *Proceedings of CSCE86,* Dec 1986, pp 229-246.

Kreifelts, T., Licht, U., Seuffert, P., Woetzel, G. (1984), Domino: A system for the specification and automation of cooperative office processes. in: *EUROMICRO.,* 1984, pp.33-41.

Kreifelts, T., Pankoke-Babatz, U., Victor, F. (1991): A Model for the Coordination of Cooperative Activities. in: *Proceedings of the International Workshop on CSCW,* Berlin, 1991, pp.85-100.

Kreifelts, T., Woetzel, W. (1987), Distribution and Error Handling in an Office Procedure System., in D.Tsichritzis and G. Bracchi (eds.):*Office Systems: Methods and Tools*. Elsevier Science Publishers B.V., North-Holland, pp.197-209.

Kumon S., (1992), *From wealth to wisdom: A change in the social paradigm*. Proceedings of CSCW 1992, Toronto, November 1992, p 3.

Kunii, T.L. (ed.),Three Dimensional Software Visualization: A Framework and Its Applications, *Visual Computing*, 1992.

Lantz, K. A. (1986), An experiment in integrated multimedia conferencing, in *Proceedings of CSCW 86*, Austin, Texas, December 1986.

Lauwers J.C., Lantz K.A.(1990) Collaboration awareness in support of collaboration transparaency: Requirements for the next generation of shared window systems *Proceedings of CHI 1990* Seattle, Washington April1-5, 1990, ACM press, ISBN-0-201-50932-6.

Lawson, B. (1980). *How designers think*. Chatham: W & J Mackay Ltd.

Lee, J. (1990). SIBYL: A tool for sharing knowledge in group decision making. *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW 1990).*

Leevers, D., (1992), *Prototyping Multimedia Communications for Manufacturing SMEs*, Presented at CIM Europe Seminar, 2-3 July 1992.

Lefebvre, H. (1991) *The production of space*. Oxford: Blackwell.

Lochovsky, F.H., Hogg, J.S., Weiser, S.P., Mendelzon, A.O. (1988), OTM: Specifying Office Tasks. in (eds.): *ACM Proc. Conference on Office Information Systems,* Palo Alto, CA, 1988, pp.46-54.

Lougher.R. and Rodden T (1993) Supporting Long term collaboration in software maintenance, *proceedings of COCS93 ACM international conference on Organisational Computing Sysstems,* San Jose, October 1993, ACM press.

Lövstrand, L. (1991), Being selectively aware with the Khronika System, in L. Bannon, M. Robinson and K. Schmidt (eds.):*ECSCW 1991. Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, Kluwer Academic Publishers, Amsterdam, 1991, pp. 265-279.

Lynch, K., (1960)*The Image of the City*, M.I.T Press, Cambridge, MA, 1960.

Lynch, M. (1985), Discipline and the material form of images: an analysis of scientific visibility, *Social Studies of Science,* 15, pp. 37-66.

Malone T.W., Lai K., Fry C. (1992), Experiments with Oval: A Radically Tailorable Tool for Cooperative Work. *Proceedings of CSCW92: ACM 1992 conference on Computer Supported Cooperative Work,* October 31 to November 4, 1992, Toronto, Canada, ACM press, pp. 289-297.

Malone, T.W., Grant, K.R., Lai, K., Rao, R., Rosenblitt, D. (1986): Semi-Structured Messages are Surprisingly Useful for Computer-Supported Coordination. in: *Proceedings of conference on CSCW86,* Austin, Texas, 1986, pp.102-114.

Mantei, M., Baecker, R., Sellen, A., Buxton, W., Milligan, T., (1991), Experiences in the Use of a Media Space, *Proceedings of CHI '91,* ACM Press.

Mariani J. & Prinz W. (1993), *Awareness in Collaborative Object Systems*. COMIC Project Report GMD-4-3, January 1993.

Mariani J. Rodden T. (1991), The impact of CSCW on Database systems, *IFIP international workshop on CSCW*, Informatik informationen-reporte, 4/1991,.K. Gorling and C. Sattler (Eds), Berlin, April 9-11 1991, pp 146-162.

Mariani, J. Lougher, R., (1992), TripleSpace: an Experiment in a 3D Graphical Interface to a Binary Relational Database, *Interacting with Computers,* Vol 4, No. 2, 1992, pp147-162.

Marmolin H.& Sundblad Y. (1993), *Sharing knowledge in a distributed environment for collaboration*. COMIC Project Report SICS_KTH-4-1, June 1993.

Marmolin H., Sundblad Y. & Pehrson B. (1991), An Analysis of Design and Collaboration in a Distributed Environment. *Proceedings of ECSCW91,* Amsterdam, September 1991, pp 147-161.

Marmolin H., Sundblad Y., Tollmar K., Avatare A. & Eriksson H. (1992a). CoDesk - An interface to TheKnowledgeNet, Design and Implementation. *Proceedings of the 4th MultiG Workshop,* Stockholm, May 1992

Marmolin H., Sundblad Y., Tollmar K., Avatare A., Eriksson H. (1992b), TheKnowledgeNet with the collaborative desktop interface. Formal demonstration *Proceedings of CSCW92,* Toronto, November 1992, p 13.

Marmolin, Sunblad, Lidbaum, Tobiasson, *The Collaborative Desktop: Interface and Functionalities*, Proceedings of the 3rd Multi-G workshop, Stockholm, December 17th 1991, pp. 17-31.

Marx, K. (1857/1970) *Capital: Volume 1*. London: Lawrence Wishart.

Massey, D. (1984) *Spatial divisions of labour: Social structures and the geography of production*. London: Methuen.

Medina-Mora, R., Winograd, T., Flores, R., Flores, F. (1992), The Action Workflow Approach to Workflow Management Technology. in *Proceedings of the conference on CSCW92,* Toronto, Canada. November, 1992, pp.281-288.

Mockapetris, P., (1984), The Domain Name System, in *Computer Based Message Services (Proc. of the IFIP 6.5 Conference),* Nottingham, UK, 1984, Hugh Smith (ed), North-Holland.

Moran T. & Andersson R (1990) The Workaday World as a Paradigm for CSCW Design. In *In proceedings of CSCW 1990,* Los Angeles California, October 1990, pp 381-393.

Natanson, M. (1986), *Anonymity,* Bloomington, Indiana University Press.

Navarro, Medina, Rodden, Environment Support For Cooperative Working, *Proc. 1992 IFIP International Conference on Upper Layer Protocols, Architectures and Applications,* May 27-29 1992, Vancouver, North Holland, pp. 69-83

Nelson T.(1980). *Replacing the printed word. A complete literary system*. Proceedings of the IFIP Congress 1980, S.H. Lavington (ed.), 1980, North-Holland, 1013-1023.

Nielsen, J.,*Hypertext and Hypermedia*, 1990, Academic Press, San Diego

Olsen, K.A., Korfhage, R.R., Sochats, K.M., Spring, M. and Williams, J., (1993), Visualisation of a Document Collection: The VIBE System, *Information Processing and Management,* Vol. 29, No 1, pp. 69-81, Pergamon Press Ltd, 1993.

OMG (1991) - The Common Object Request Broker: Architecture and Specification, OMG Document Number 1991.12.1, Revision 1.1, OMG, Draft 0, December 1991.

Orlikowski, W. J. (1992). Learning from Notes: organizational issues in groupware implementation. In J. Turner & R. Kraut (Ed.), *CSCW 1992,* (pp. 362-369). Toronto: ACM Press.

OSF, *Distributed Computing Environment: An Overview*, Available from the OSF January 1992.

Patterson, J. F., Hill, R. D., Rohall, S. L., Meeks, W. S., (1990) Rendezvous: An architecture for synchronous multi-user applications, in *Proceedings of CSCW 1990*, October 7-10, Los Angeles, Ca., ACM, 1990, pp.317-328.

Patterson, J.F. (1991), Comparing the Programming demands of Single-User and Multi-User applications.*User Interface Software and Technology (UIST),* November 11-13, 1991, pp.87-95.

Pehrson, B., Gunningberg, P. and Pink, S. MultiG-A research Programme on Distributed MultiMedia Applications and Gigabits Networks, *IEEE Network Magazine* vol 6, 1 (January 1992), pp. 26-35.

Pollner, M. (1987), *Mundane Reasoning,* Cambridge, Cambridge University Press

Prinz W. (1993) TOSCA: Providing Organisational Information to CSCW Applications *Proceedings of ECSCW 1993, Second European Coonference on CSCW,* Milan, September 1993.

Raffel, S. (1979), *Matters of Fact,* London, Routledge and Kegan Paul.

Rein, G. L., Ellis, C. A., (1991) rIBIS: A real-time group hypertext system, *International Journal of Man Machine Studies*, 34(3), March 1991, pp 349-368.

Robinson M. (1993) Keyracks & Computers, To appear in *Wirtschaftsinformatic* 2/93, (1993).

Robinson, M. (1993). Design for Unanticipated Use, *Proc. ECSCW93.*

Rodden T., (1992a) ECSCW91 Developers Workshops, *ACM SIGOIS Bulletin*, Vol 5, No 3, Summer 1992, ACM Press.

Rodden T., Mariani J., Blair G. (1992): Supporting Cooperative Applications, *CSCW: An international Journal,* vol. 1, no. 1, Oct. 1992, Kluwer

Rodden, T. Sommerville I, Twidale M., (1993), The DNP: Understanding and supporting the work of Designers, *Proc. ECSCW93,* Milano, Sept 1993, Kluwer.

Rodden, T., Blair, G.S. (1991): CSCW and Distributed Systems: The Problem of Control. in L. Bannon, M. Robinson and K. Schmidt (eds.):*ECSCW 1991. Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, Kluwer Academic Publishers, Amsterdam.

Root, W. R. (1988). Design of a multi-media vehicle for social browsing. In *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW 1988)*, (pp. 25-38). Portland, Oregon: ACM Press.

Rozier, M., Abrossimov, V., Armand, F., Boule, B., Gien, M., Guillemont, M., Herrmann, F., Kaiser, C., Langlois, S., Leonard, P., Neuhauser, W. (1990): Overview of the CHORUS Distributed Operating Systems, CS/TR-90-25, Chorus Systemes. April 15.

Rudebusch, T.D. (1991), Supporting Interactions within Distributed Teams. in: *International Workshop on CSCW,* April 9-11, 1991, pp.17-33.

Rumelhart, D. E., & Norman, D. A. (1981). Analogical processes in learning. In J. R. Anderson (Eds.), *Cognitive Skills and their Acquisition* (pp. 335-360). Hillsdale, NJ.: Erlbaum.

Sacks, H. (1963), *Sociological Description,* Berkeley Journal of Sociology, 8, pp. 1-16.

Sarin, S. and Greig I., (1985), Computer-Based Real-Time Conferencing Systems, in *IEEE Computer*, Vol 18, No 10, (October 1985), pp. 33-45.

Schmidt, K. (1991): Riding a Tiger, or Computer Supported Cooperative Work, in L. Bannon, M. Robinson and K. Schmidt (eds.):*ECSCW 1991. Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, Kluwer Academic Publishers, Amsterdam.

Schmidt, K. and Rodden, T. (1992), Putting it all together: Requirements for a CSCW platform, Department of Computing, Lancaster University.

Shapiro, M. (1986): Structure and encapsulation in distributed systems: The proxy principle. in:*6th International Conference on Distributed Computer Systems,* May, 1986, pp.198-204.

Sharrock, W.W. and Anderson, R.J. (1986), *The Ethnomethodologists,* London, Tavistock.

Sharrock, W.W. and Anderson, R.J. (1987), Work flow in a paediatric clinic, in Button, G and Lee, J. (eds), *Talk and Social Organisation,* Clevedon, Multilingual Matters, pp. 244-260.

Shu and Flowers, (1992), Groupware Experiences in Three-Diemensional Computer-Aided Design, in *Proc. CSCW92,* ACM Press, 1992.

Sloman, M., Moffett, J. (1989): Managing Distributed Systems, Domino Project Draft Report (19 September 1989), Imperial College, London.

Smalltalk 1992, ParcPlace VisualWorks\Smalltalk-80, ver 1.0, 1992

Smith, R., Experiences With The Alternate Reality Kit: An Example of the Tension Between Literalism and Magic, in *Proceedings of CHI + GI87, Human Factors in Computing Systems IV*, Toronto, 5-9 April 1987, North Holland

Ståhl, O., (1992), Implementation Issues of Aura Based Tools, In *Proc. 5th MultiG Workshop,* Stockholm, December 18, 1992.

Ståhl, O.,Tools for Cooperative Work in the MultiG TelePresence Environment, *Proceedings of 4th Multi-G Workshop*, May 12th 1992, Stockholm-Kista, BjÖrn Pehrson (ed), pp. 75-88.

Stefik, M., Bobrow, D.G., Foster, G., Lanning, S., and Tatar, D.(1987b), WYSIWIS Revised: Early Experiences with Multi-User Interfaces, *ACM Transactions on Office Information Systems,* Vol. 5, No. 2 (April 1987), pp 147-167.

Stefik, M., Foster, G., Bobrow, D.G., Kahn, K., Lanning, S., and Suchman, L.(1987a), Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings, *Communications of the ACM*, 30(1), January 1987, pp 32-47.

Suchman, L.A. (1983): Office Procedures as Practical Action: Models of Work and System Design, *ACM Transactions on Office Information Systems*, vol.1, no.4, October 1983, pp. 320-328.

Sundblad Y., Marmolin H. & Tollmar K (1992). The Collaborative Desktop: Video 12 min, IPLab, NADA, October 1992.

Takemure, H. and Kishino, F., (1992), Cooperative Work Environment Using Virtual Workspace, In *Proc. CSCW92,* Toronto, Nov 1992, ACM Press.

Tang, J. C. and S. L. Minneman. (1990). Videodraw: A video interface for collaborative drawing. *ACM SIGCHI Conference on Human Factors in Computing Systems*. 313-320.

Tang, J.C., & Minneman, S.L. (1991). VideoWhiteBoard: video shadow to support remote collaboration. In *CHI 1991,* New Orleans.

Tansel, A., Clifford, J., Gadia, S., Jajodia, A., Segev, A. and Snodgrass, R. (eds), (1993), *Temporal Databases: Theory, Design, and Implementation, Database Systems and Applications Series* Benjamin Cummings Pub. CO., Redwood City, CA, January 1993.

Tepper A., (1992), Future Assessment by Methaphors. Paper presented at the Post-Congress Workshop Methods in *Technology Assessment and Decision Making of the 3rd European Congress on Technology Assessment (ECTA III),* Copenhagen, 4 -7 November 1992.

The Object Management Group (OMG) (1990), Object Management Architecture Guide, OMG Document Number 1990.5.4, Available through the Object Management Group, 1990.

The Object Management Group (OMG), *Object Management Architecture Guide*, OMG Document Number 1990.5.4, Available through the Object Management Group, 1990.

Trevor, J.J., Rodden, T., Blair, G.S. (1993): COLA: A Lightweight Platform for CSCW. in (eds.): *Proceedings of ECSCW93,* Milan, 1993.

Trist, E. (1971), Critique of scientific management in terms of socio-technical theory, *Prakseolgia,* vol. 49-50, pp. 159-174

Twidale, M., *Rules of Etiquette at the global cocktail party: issues of metaphor in CSCW design*, COMIC Working Paper, COMIC-LANCS-4-3, 1993.

Van der Linden, R.J., Naming and Federation, in *Naming Facilities in Internet Environments and Distributed Systems* (Special Issue of the Distributed Processing Technical Committee Newsletter), Bharat Bhargava (ed), Vol. 14, No 1, pp 21-27, IEEE Technical Committee on Distributed Processing, June 1992.

Weber, M. (1978), *Economy and Society,* vols 1 and 2. Berkeley, University of California Press.

Whittaker, S., Brennan, S. E., & Clark, H. H. (1991). Coordinating activity: An analysis of interaction in computer supported cooperative work. *In SIGCHI Conf on Human Factors in Computing Systems*, (pp. 361-368), New Orleans, ACM Press.

Wolf, C. G., Rhyne, J. R., & Briggs, L. K. (1992). Communication and information retrieval with a pen-based meeting support tool. In J. Turner & R. Kraut (Ed.), *CSCW 1992*, (pp. 322-329). Toronto: ACM Press.

Yoder, E., R. Akscyn and D. McCracken. (1989). Collaboration in KMS, a shared hypermedia system . *Proceedings of the SIGCHI Human Factors in Computing Systems*. 37-42.

Young, F.W., and P. Rheingans, Visualising structure in high-dimensional multivariate data, *IBM Journal of Research and Development,* Vol. 35, No. 1/2, January/March 1991, pp. 97-107.