A Survey of CSCW Systems

Tom Rodden
Department of Computer Science
Lancaster University
Bailrigg, Lancaster
LA1 4YR.
Email: tom@comp.lancs.ac.uk

Abstract

Over the last decade, Computer Supported Cooperative Work (CSCW) has emerged as an identifiable research area which focuses on the role of the computer in group work. CSCW is a generic term which combines the understanding of the nature of group working with the enabling technologies of computer networking, systems support and applications. This paper examines the classes of system which have emerged to support cooperative working. A framework for characterising and describing CSCW systems is presented and four major classes of cooperative system identified. Each of these classes of cooperative system are examined highlighting their general characteristics and applicability to CSCW.

Contents

1. Introduction

The advent of powerful low cost personal computing has ensured that computer systems and tools are available which help users perform the tasks associated with their profession. Such systems have penetrated large segments of normal work practice and personal access to computer systems is no longer unusual. However, most of these computer systems and tools have often been considered in isolation, both of other tools and the existence of other people using similar computer based tools. Given both the widespread availability of modern networking technology and the reliance of the success of most projects on the cooperative activities of people, this historical focus has limited the support provided by automated tools.

Over the last decade, Computer Supported Cooperative Work (CSCW) has emerged as an identifiable research area which focuses on the role of the computer in group work. The research being undertaken poses a number of questions. How can computers be exploited to maximise the synergy of groups?; What kinds of software should be developed?; How do we define group work? To address these problems CSCW involves researchers across a range of disciplines including psychology, sociology, organisational theory, and anthropology. Bannon [Bannon 91] highlights the roles and issues addressed by each of the different disciplines involved in CSCW. He maintains that research will require a multi-disciplinary approach. Within that context this paper examines the technology involved in CSCW, highlighting the different classes of system which have emerged to support group working.

The following section describes a framework for characterising and describing CSCW systems and outlines four major classes of cooperative system. The remainder of the paper examines each of the classes of cooperative support system outlined in section 2 in detail. The range of systems based on messaging technology are considered in section 3. Section 5 deals with computer conferencing systems while support for face-to-face meetings is discussed in section 5. Finally, section 6 considers the development of co-authoring and argumentation systems.

2. A classification of CSCW systems

An essential precursor to the study of collaborative systems is the definition of a mechanism for classifying these systems. Two principal characteristics are common to all cooperative systems and form the basis for the classification used in this paper.

1. THE FORM OF INTERACTION (SYNCHRONOUS VERSUS ASYNCHRONOUS)

Creative problems, such as those tackled by brain-storming, require group members to cooperate in a *synchronous* manner since the creative input of each group member is required to generate a strategy for tackling the task. In contrast, prescriptive tasks have a previously formulated solution strategy where group members take on particular roles and work in an *asynchronous* manner often without the presence of other group members. Cooperative systems are therefore either synchronous or asynchronous systems.

2. THE GEOGRAPHICAL NATURE OF THE USERS (REMOTE VERSUS CO-LOCATED)

Computer support for group interaction has traditionally considered the case of geographically distributed groups who work asynchronously with each other. More recent research is aimed at the support of face to face meetings. Using this classification cooperative systems are either *remote* or *co-located.* This division is as much logical as physical and is concerned with the accessibility of users to each other rather than their absolute physical proximity. The term co-located is used to emphasis this logical division and to avoid confusion with the distinction between remote and local communication systems.

2.1 The classes of CSCW systems

Four classes of cooperative system have emerged over the last decade (Figure 1).

Message systems

Message systems are the most mature class of system. They have evolved from electronic mail programs which allowed a user using a central machine to send textual messages to other users on the same machine. As wide area networks designed to support computer communication became more widespread [Mortensen 85], electronic mail systems increased in complexity and functionality.

This development resulted in the formation of a number of different standards for electronic message systems. The most recent being the Message Handling System (MHS) model described in the CCITT X400 series of standards documents [CCITT 87]. Message standards normally describe the message format used to transfer information in message systems. Structured message systems are based on the principle of extending the amount of machine processible semantic information available by adding syntactic structure to the existing message structures.

Computer conferencing

Computer conferencing systems are also related to electronic mail programs. However, the principles are different in that structure is imposed in terms of how messages are grouped. A typical computer conferencing system consists of a number of groups (called conferences), each of which has a set of members and a sequence of messages. Conferences are often arranged so that they individually address a single topic and users subscribe to conferences of interest . Usually, the system stores information about how far every member has read in each conference. This information is normally held with conference messages within one central database rather than the individual mailbox approach used in messaging systems.

The development of reliable high speed communications has lead to the emergence of new *real-time conferencing* systems. These allow conference members to communicate in real-time. *Multi-media conferencing* systems represents the introduction of a new technological development into conferencing systems. As computer systems become more powerful their capability to handle wider classes of data increases. This has led to multi-media systems which integrate audio, text and video.

Meeting rooms

A typical automated meeting room consists of a conference room furnished with a large screen video projector, a computer (or network of computers), video terminals, a number of individual input/voting terminals, and a control terminal. The computer system supporting the meeting often makes use of multi-user software based on some form of analytical decision technique. Software for graphics and vote tally and display also form part of the normal meeting facility.

Co-Authoring and Argumentation Systems

Co-Authoring and argumentation systems are a general class of system which aim to support and represent the negotiation and argumentation involved in group working. The cooperative authoring of documents is demonstrative of this class of cooperation where the final generation of a document represents the product of a process of negotiation between authors.

The form of interaction and the geographical nature of cooperative systems are independent and can be considered as orthogonal to each other. The resultant classification space is shown in figure 1 with the classes of systems highlighted previously placed within this classification space.

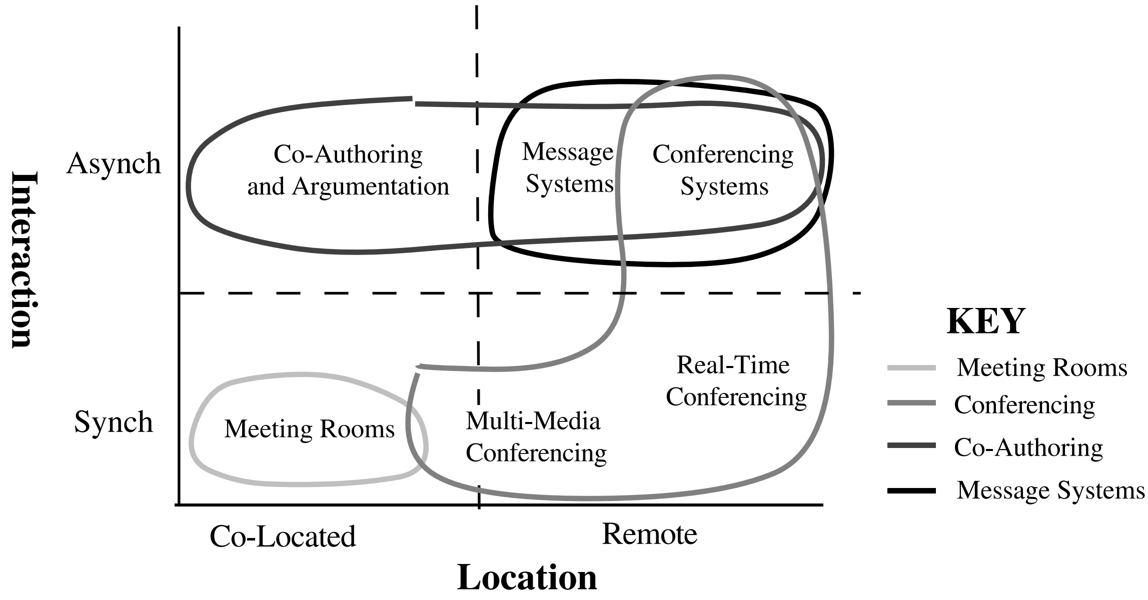

Figure 1: A Classification space for CSCW systems

This classification space shown in figure 1 shows how CSCW system can be classified depending on two major characteristics. A further characteristic common to all cooperative systems is control [Rodden 91]. Two predominant control mechanisms have emerged within CSCW systems, speech act theory systems and procedure based systems. Both these forms of control are discussed in more depth in appendix A.

3. Message systems

Cooperative message systems are often termed structured or active message systems and assume an asynchronous and remote mode of cooperation. The assumption underlying these systems is that members of a group cooperate by exchanging messages.

Because of their history, structured message systems are based upon concepts identified in electronic mail systems and a standard message interchange format is often assumed. Cooperative message systems attempt to augment the structure of these standard message organisations to extend the amount of machine processible information available within the system.

The object model is often used to augment the structure of the messages and these systems employ a model of users exchanging message objects. Each message object has attributes set to appropriate values for the message being sent.

Structured message systems either use a formal model of control based on one of the control models highlighted in appendix A or use a philosophy of semi-automation which adopts a more pragmatic model of control providing support where appropriate. A number of systems using formal models of control augment and develop the two major models of control highlighted in appendix A.

Message systems also differ in the mechanisms used to represent control . The primary means has either been a procedural language used in office automation systems or the network based approach adopted by systems based on speech act theory. Figure 2 lists a range of research projects using message based techniques and the form of control adopted by the various systems

Research Project	Control Model	Representation
Coordinator [Winograd 87]	Formal Speech Act	Network
Information Lens [Malone 87]	Semi-Formal	Production Rules
Chaos [De Cindio 86]	Formal Speech Act	Network
Domino [Kreifelts 86]	Formal Procedural System	Script Based
Cosmos [Wilbur 88]	Formal Augmented System	Script Based
Amigo [Danielson 86]	Formal Augmented System	Script Based
Strudel [Sheperd 90]	Semi-Formal	Production Rules

Figure 2 The control models used by structured message systems

This section will describe in detail two significant examples of structured message systems, one uses a formal model to represent control while the other adopts a semi-formal approach. The COSMOS project is presented because of its philosophy of constructing formal models of the cooperation involved and because it augments the traditional models of control. This approach contrasts with the pragmatic approach adopted by the Information Lens described in section 3.2

3.1 A Formal Approach- The COSMOS Project

The COSMOS project was a UK industry/university project to specify, design and prototype a configurable message system to support structured group working.

Group communication within COSMOS is based on the use of Communication Structures (CS). In COSMOS, a communication structure is a set of rules which characterise the form of a communication activity. Since this is an abstraction, communication acts are described as the exchange of message objects between agents enacting specific roles. For example a CS might describe committee structure including roles (chair-person, secretary, participant), message objects (minutes, agendas, etc..) and rules (procedures) relating them. Communication structures are defined using a structure definition language (SDL) [Churcher 88].

A CS definition is used to instantiate a Cosmos *Activity*. This may be achieved by a user selecting the CS that embodies the required type of group activity , and then causing an instantiation to be created by supplying the data needed to execute it. An identified activity is performed by each user carrying out the actions for his/her role, subject to specific constraints and the rules and conditions specified in the CS. Messages are exchanged with other roles within that activity, although personal mail may be sent at any time. The user is able to ascertain what action is required next and possibly which other users are involved in the activity.

The Cosmos project has demonstrated the rationale for a structure description language which supplements the basic concepts of speech act theory to describe local and global management systems for inter-role cooperation [Bowers 88].

3.1.1 Control Representation

The COSMOS project is more formal in its handling of group working and roles than the Information Lens which is considered in the following section. Within COSMOS this formality is represented as *communication structures*.

Communication structures, roles, message objects, actions, rules, and activities are the principal elements of the COSMOS framework for group working and form the basis of a conceptual framework for SDL. *Actions* may be either encapsulated actions or exchanges. An *encapsulated action* involves the creation of some message object which is typically exchanged on completion. Exchanges are the elementary communicative acts, and involve at least two roles and, typically, one message object.

An exchange involves a *messaging act* in which control over access to one or more message objects is transferred between role, and an optional *illuctionary act* which establishes relations of local constraints with other exchanges (for example, a question would render an answer expectable). *Roles* are the agents and patients of actions, they may correspond to people, a group or an automaton. *Rules* appear as either production rules, if a condition must be met, or as context free rewrite rules.

Conditions can involve either states or events within the system. *Temporal order constraints* constrain the order in which events may occur and are expressed as order constrained statements. These constraints may be deterministic, e.g. A <B (A must occur before B), non-deterministic e.g., A,B (A and B can occur in any order) or a combination of both e.g., A<B, C. The interrelation of these various components within the SDL are shown in figure 3

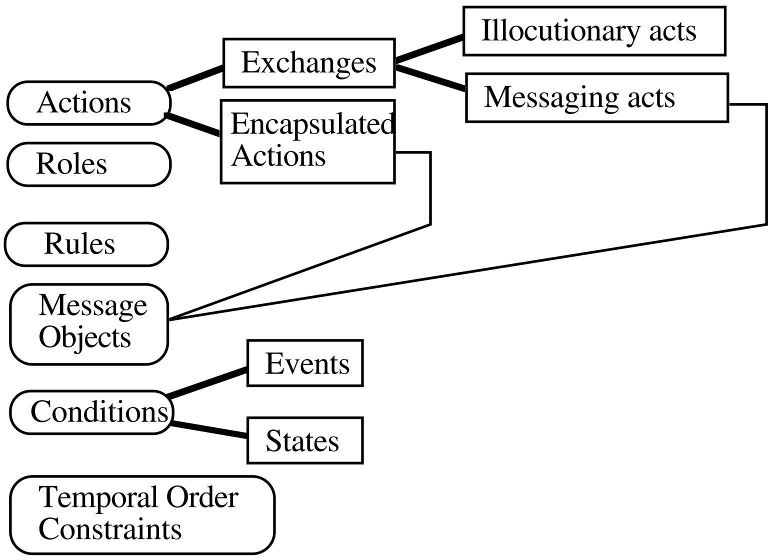

Figure 3 A conceptual framework for SDL

An illustrative example of SDL given by the developers of COSMOS involves the development of a communication structure for voting (figure 4). A communication structure consists of a name, a set of rules and a set of order constrained statements.

```
VOTINGCS

  VOTINGCS -> OPEN VOTE(n) CLOSE
  /* Voting has three phrases                                         */
  /* The Open phase creates a participant role which makes messages   */
  /* visible to all roles. A call for vote message is created by the chair */
  /* role. The call for vote message is then sent to participants . The */
  /* proposer role then creates a proposition message which is sent   */
  /* to participants.                                                 */

  OPEN -> allocate Chair Proposer Teller Voter .. to Participant
          Chair create call_for_vote
          call_for_vote Chair Participant
          Proposer create proposition
          proposition Proposer Participant

  /* The Voting phase voter creates a vote message which is then      */
  /* sent to the participant role                                     */
  VOTE(n)-> can ( Voter(n) create vote(n) from proposition
          & vote(n) Voter(n) Participant)

  /* The Close phase teller creates a new message from the vote       */
  /* messages and sends this message to the Participant role          */

  CLOSE        -> Teller create announcement_of_results from vote ..
          announcement_of_results Teller Participant
          call_for_vote Chair Participant < vote(n) Voter(n) Participant
```

Figure 4 Structure Definition for a Voting Communication Structure

The process of voting is represented as having three phases; an opening phase, the actual voting and a closing phase. There are four obvious roles, plus a fifth *participant* which is introduced to represent what the others have in common, e.g., they can all see every vote. There are four kinds of message objects: *call_for_vote* , *proposition* , *vote(n)* and *announcement_of_results* . Exchanges are expressed by specifying a kind of message object followed by a sender role and a receiver role.

3.2 An Informal Approach- The Information Lens

The COSMOS approach is representative of research being undertaken by the European message handling community within the field of CSCW. The Information Lens on the other hand represents a strand of research which aims at merging traditional message handling with classical A.I. techniques. Malone's primary concern is the development of information sharing systems which minimise the effects of information overload. As Hiltz and Turoff [Hiltz 85] argue, effective systems must give message recipients the ability to discriminate between those messages they wish to read and those of little relevance to them. This is often achieved by imposing some structure on the set of messages a user receives [Palme 84], so that messages are categorised to allow the user to select those of interest.

As a result of several studies on how information is shared in organisations [Brobst 86], Malone developed the Information Lens. The Information Lens system emphasises what Malone terms a cognitive approach to information sharing. Malone describes *cognitive filtering* as being the filtering of information based upon the topic of the message. The system utilises a number of techniques from artificial intelligence including frames, production rules and inheritance.

A number of key ideas form the basis of the Information Lens.

- Semi-structured message types (or frames) form the basis for an intelligent information system.
- Sets of production rules can be used to specify automatic processing of messages.
- A consistent set of display oriented editors should be provided.
- Frame inheritance lattices allow the specialisation of message types.

3.2.1 Representing Control

In contrast to the approach of COSMOS, the Information Lens follows a more pragmatic semi-formal approach to group working by attempting to support it through controlled support for automated distribution. Messages addressed to *anyone* as an addressee will be delivered via an automatic mail sorter which runs on a workstation and periodically retrieves mail from a special mailbox(figure 5).

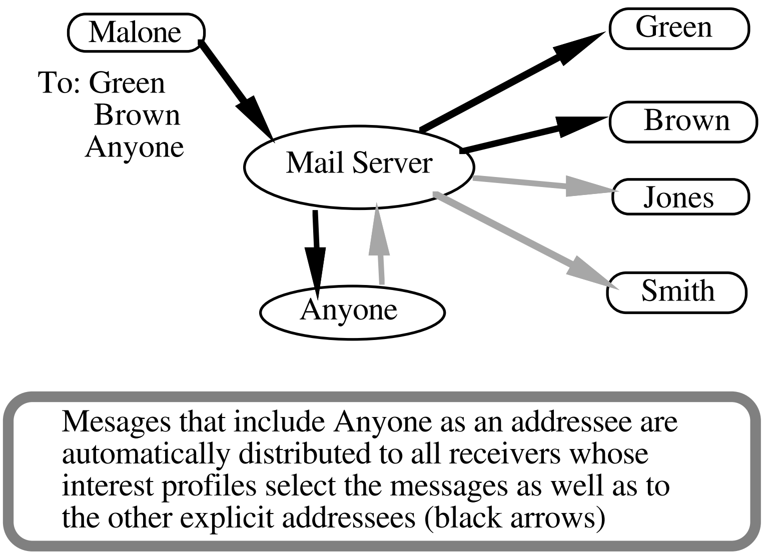

Figure 5 The logical arrangement of the Information Lens

For each message type, the system includes a template with a number of fields for holding information. Associated with each field are several properties including the default value, a list of possible alternative values, and a comment explaining the field. Users amend the values of fields within a template by the use of a display-orientated editor (figure 6). The message templates are arranged in a inheritance hierarchy. The more general message types are at the top of the hierarchy and the more specific types at the base of the hierarchy, inheriting properties from those above them.

Figure 6 A Lens Message Template

The Lens Environment allows users to construct rules for finding, filtering, and storing messages. Rules consist of a test and an action; if a message satisfies the test, the action specified by the rule is performed on the message. A display-oriented editor is used for the construction of rules (figure 7).

Figure 7 The Lens Rule Editor

Construction of the IF part of a rule involves filling in selection specifications for the different message fields. Specifications consists of number of tests on the contents of a message field combined using *and*, *or* and *not* operators. If specifications appear in more than one field, then all specifications must be satisfied for the rule to succeed. The THEN part of rule is chosen from one of a number of message handling primitives including move, delete, save, reply etc.

Malone has subsequently enhanced the work on the Information Lens to produce a more general system called the Object Lens [Malone 88]. The Object Lens has generalised the message form objects of the Information Lens to allow the construction of arbitrary objects within an *object store*. In addition to the specialisation found in the Information Lens users can construct general relationships between objects through the use of *object links*. Malone argues that this capability is akin to the facilities provided by current hypertext systems.

The Object Lens allows the capture of active rules as a set of *semi-autonomous agents.* Semi-autonomous agents within the Object Lens are objects which consist of a set of rules which are obeyed when an agent is *triggered*. Agents can be triggered by a events such as the arrival of new mail, at a particular time or explicitly by another agent. Malone views the Object Lens as a cooperative application generator system embodying many of the principles found in hypertext, object-oriented systems and rule-based agents. The system has successfully being used to generate a range of CSCW applications including Answer Garden [Ackerman 90] a system for holding organisational expertise and SIBYL [Lee 90] a system for representing design rationale.

4. Conferencing systems

Although sharing a great deal of common experience with electronic message systems, computer conferencing systems developed independently and were first envisaged in the early 1970's. During the Nixon administration, the Office of Emergency Preparedness (OEP) in the USA commissioned Murray Turoff to create a computerised version of the tele-conferencing facilities in use at that time. Turoff responded by developing the **E**mergency **M**anagement **I**nformation **S**ystem **A**nd **R**eference **I**ndex (EMISARI)

The EMISARI system [Hiltz 78] operated as an electronic network linking the ten OEP regional offices and eliminated the constraints of time and geographic location. EMISARI consisted of two systems: *Party - Line*, the computerised counterpart of the telephone conference call, and *Discussion* an on-line file cabinet of topic specific messages stored on line for all to see and comment on.

These two elements are the building blocks for current computer conferencing systems. In computer conferencing systems users interact through a shared information space accessed by each of the users as shown in figure 8.This model of interaction through a shared information space is often augmented by the use of direct user to user communication.

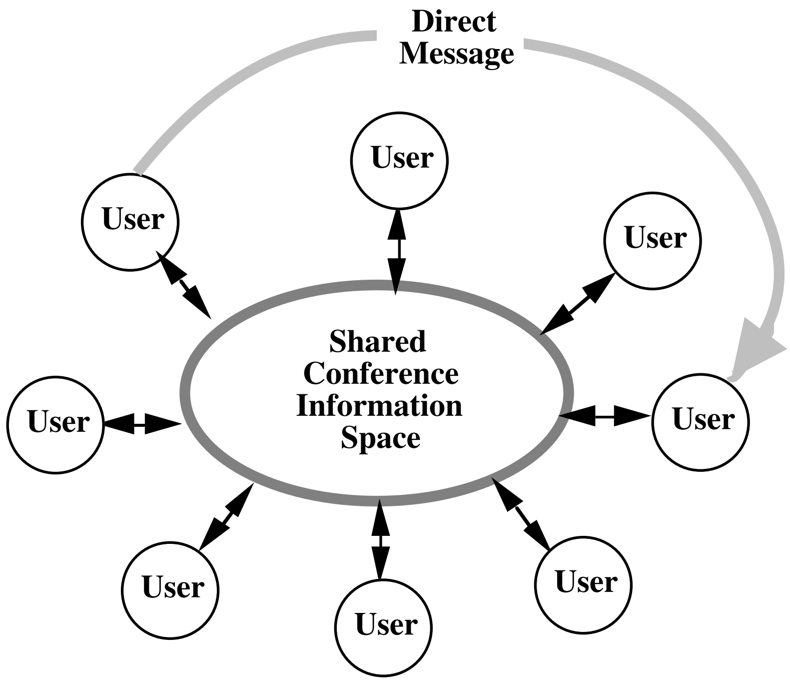

Figure 8 Interaction in Conferencing Systems

The term computer conferencing has been used to signify a wide range of cooperative systems. These different types of conferencing system have emerged depending on two principal features of the shared information space:

1. *The form of interaction with the shared information space*
   User can interact with the shared conference space either asynchronously over a long time period or synchronously in real-time.
2. *The type of information represented in the information space*
   Many forms of information may be represented within the shared information space. In the past this information has been textual, but with the advent of modern workstation capabilities an increasing range of media can be represented.

The development of these facets of the shared information space has led to the emergence of a number of distinct forms of computer conferencing systems. The oldest form of conferencing systems provided only asynchronous sharing of textual information. Many of these *Traditional ConferencingSystems* are still in use today. The emergence of high bandwidth local area networks resulted in the development of *Real-Time Conferencing Systems* which support real-time concurrent access to the shared information space. The recent predominance of workstation technology and the integration of different media within workstations has led to the development of *Multi-media Desktop Conferencing Systems*.

The different forms of conferencing systems are examined in this section by considering examplars of each of the major varieties of conferencing system. Section 4.1 considers traditional conferencing systems while section 4.2 highlights RTCAL, a significant example of realtime conferencing. Section 4.3 considers multi-media conferencing and concludes by discussing desktop conferencing systems showing how this class of systems is beginning to evolve the general conferencing model presented in figure 8.

4.1 Traditional Conferencing Systems

A typical computer conferencing system consists of a number of groups called conferences, each of which has a set of members and a sequence of messages. Conferences are usually arranged so that individual conferences addresses a single topic. A user subscribes to those conferences which are of interest to him. Usually the system stores information about how far every member has read in each conference. This makes it possible for the system to tell users which messages are new to them when they connect to the system.

For example, a user may subscribe to conferences on computers, information processing and music. When the user initiates a conference session, he is presented with a list of the conferences to which he belongs, and the number of messages in each, for example:-

12 Unseen entries in Computers
9 Unseen entries in Information Processing
6 Unseen entries in Music

Any replies to a message within a conference are delivered to other users belonging to that conference. Each conference is controlled by a conference leader who organises and administers general conference book-keeping. The use of conferencing structures is analogous to participating in several simultaneous conversations, each conversation having its own pre-defined topic.

Traditional computer-conferencing systems are an extension of bulletin boards which have developed from early electronic mail systems. Bulletin board systems typically offer a fixed set of topics, which can be adjusted only by system administrators. Also, new comments are simply appended to the end of the topic, with no facility to link related comments.

Conferencing systems offer more flexibility in generating new conferences and organising comments within a conference. The system provides a mechanism to learn about conferences, which allows people to interact with others who have common interests and experiences. Most conferencing systems offer facilities for monitoring changes to conferences and for searching conference comments according to criteria such as date, author, and keyword.

A large number of conferencing systems exist each with different properties and peculiarities:

- EIES, Descended from Murray Turroff's earlier EMISARI System. One of the first conferencing systems.
- PARTI, PARTIcipate conferencing system developed and marked by Participation Systems Inc. One of the leading conferencing systems in the USA.
- NOTEPAD, Developed in 1979 by InfoMEDIA Corp; NOTEPAD has been used as (among other applications) the basis for an electronic journal system [Pullinger 86].
- COM, Developed at the Swedish National Defense Research Institute in 1977. COM is in regular use in Europe and has even spun off a "portable" system called PortaCOM. Its instantiation as Eurokom serves as the principal mail and conferencing system for Esprit projects.
- EFORUM , Developed by Network Technologies International (NETI) EForum is one of the more recent conferencing systems and provides a modern workstation interface.
- CONFER, A conferencing system developed at the University of Michigan. One of the latest generation of conferencing systems offering increased flexibility over its predecessors.

As part of a move to improve communication and cooperation among geographically separate groups of engineers, Hewlett Packard in 1984 investigated computer-conferencing [Fanning 86]. After consideration of the available systems (including those above) Hewlett Packard selected and observed the use of one such system (CONFER) within its organisation. The observations derived from the Hewlett Packard experience highlight the problems associated with supporting groups using computer conferencing systems.

The most significant influences on computer conferencing were either human or environmental rather than technological. The single factor with the highest correlation with the success of an individual conference was the *activity level* of that conference's leader. Other factors which influenced a conference's success or failure included:

- Appropriateness of Discussion Topics. Chosing the remit of a conference's topics represents a considerable problem. Topics need to be sufficiently restrictive to prevent a conference wandering, but sufficiently general to include all users comments and interests.
- Overload and Pruning. Hewlett Packard found their user community produced a lot of text. As a result a new conference participant could easily spend days

reading the text of some conferences. As a result the conference systems administrators where forced to adopted harsh pruning measures (with a subsequent loss of information) in order to avoid discouraging new participants.

4.2 Real-Time Conferencing Systems

Computer conferencing systems traditionally address *asynchronous* interaction among users. However Grief [Sarin 85] outlines a number of areas, such as crisis management, where synchronous communication is necessary. Several of the current conferencing systems, such as PARTIcipate and EForum, either already provide rudimentary real time support, or are being extended to provide support for real time communication.

A prototype real-time conferencing system which is worthy of note as it highlights the principles of real time conferencing was developed by Grief at MIT. The prototype, RTCAL (Real Time CALender) supports meeting scheduling among a group of users.

RTCAL, provides computer support for the scheduling of meetings by building a *shared workspace* of information from participants' on-line calendars. The shared workspace is displayed to the users in conjunction with his/her own *personal* calendar. While RTCAL provides users with information and tools for decision support, it does not automate the selection of a meeting time.

RTCAL demonstrates a number of features applicable to most real-time conferencing systems

- Shared and private views which allow information to be visible to either a single users or the whole user community.
- The Alignment of Related Information between a users' shared and private views to be displayed appropriately.
- An on-line voting scheme lets users express their opinions on a number of proposals [1].
- Participant autonomy gives participants the ability to act independently of the user community.
- A number of conference distinct roles are provided in particular a *chairperson* overseas all activity in the conference determining who has the *floor* at any given time. Only the person given the floor has control over shared views[2].
- Presentation of status Information regarding the conference topic, it's participants, who the chairperson is; and who currently has control is displayed as a public view to all participants.
- Two types of command exist, those relating to conference control, for example requesting the floor; and those concerned with applications, for example the local editing of information.

Although RTCAL demonstrates some general principles of real-time conferencing systems it supports only a specific application activity. The development of workstation

---

[1] Like their political equivalents voting schemes within computer conferences appear to be most effective when they are anonymous.

[2] This is in contrast to the approach followed by the designers of Colab.

technology combined with high bandwidth communications has led to the emergence of a particular class of real-time conferencing system called *shared screen systems*.

Shared screen systems allow the screen contents and windows to be displayed and manipulated by more than one workstation. The development and emergence of shared screen technology has had a significant effect on synchronous cooperative systems. The concepts underlying shared screen systems have developed from the work of the CoLab [Stefik 87a] project at Xerox PARC. A discussion of the HCI principles discovered by the CoLab project is given in section 5.1

4.3 Multi-media and Desktop Conferencing Systems

The merging of workstation technology and real-time computer conferencing has had a significant impact on CSCW. This merging has been termed *desktop conferencing* and an integral part of desktop conferencing is the use of *multi-user interfaces*. Lauwers [Lauwers 90] outlines two approaches for the development of multi-user interfaces. The first approach suggested is the development of special purpose applications which are *collaboration aware*. CoLab which is described in section 5.1 is an example of a systems employing collaboration aware applications.

The second approach which can be taken is to provide facilities which allow existing single-user applications to be shared between user in a *collaboration-transparent* manner. These systems may support multi-media capabilities. Examples of work which use this approach approach include Vconf [Lantz 86], Rapport [Ahuja 88], SharedX [Gust 88], Conference Toolkit [Bonfiglio 91] , and MMConf [Crowley 90].

The most generally used model of a shared window systems is as show in figure 9. The role of the conference agent is to multiplex the output from the applications onto the users workstations and to route the input from the users to the appropriate applications.

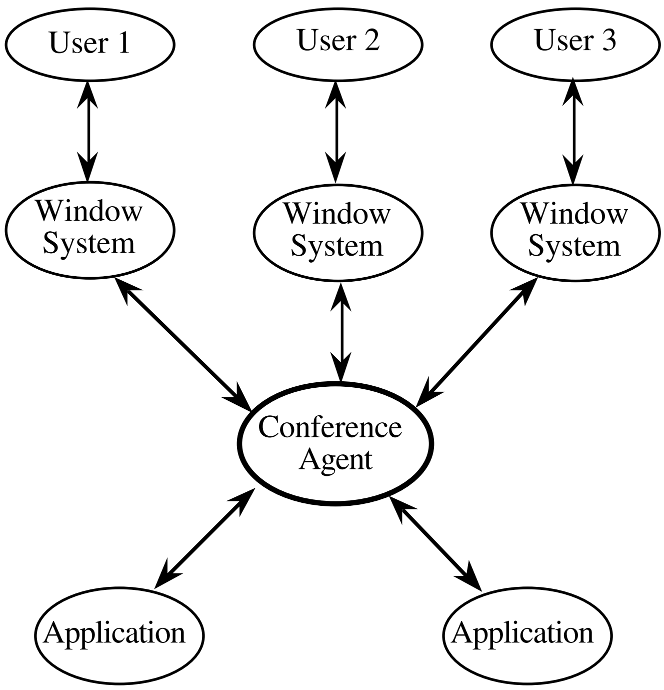

Figure 9: A shared window model

Each window system has control for the local layout and placement of windows. As well as the I/O control necessary for the sharing of applications, the conference agent is also responsible for:

- Floor Control: controlling the input from users to allow input to come from the user who is currently authorised to input for a particular application. This is analogous to passing the pen for the board in a meeting .
- Workspace Management : the layout, placement and grouping of shared windows is controlled within the shared workspace by the conference agent. The allows alterations and changes to the layout to be reflected in other users' workstations.
- Dynamic Reconfiguration: the addition of new users and the early departure of existing user is the responsibility of the conference agent. Each addition or departure necessitates the reconfiguration of the system.
- Secretarial Functions: the setting up of conferences, the initialisation of appropriate systems and the recording and logging of conferences are often provided by the conference agent.

An interesting development in computer conferencing is the emergence of systems which aim to support synchronous cooperative working in a manner which unifies both remote and co-located users. These systems combine the shared screen facilities provided within real-time conferencing systems with video and audio in a unified manner.

One example is the Rapport Multi Media Conferencing System [Ahuja 88] which is illustrative of a number of current systems which exploit multi-media and real-time facilities including MMconf at BBN and the CCWS system [Poggio 85] from SRI. All of these systems, like most interactive real-time conferencing facilities, are currently limited by technology to supporting locally remote groups. Locally remote groups are those groups which, although demonstrating many of the working practices of geographically distributed groups, exist within one organisation or site. As a result, systems which support multi-media communication and cooperation in an asynchronous manner have evolved; these include Diamond [Forsdick 82] at BBN and the Andrew Message System [Borenstein 88] at CMU.

The Rapport multimedia conferencing system was developed at AT&T Bell Laboratories in New Jersey. The current system supports interactive, real time distributed conferences among two or more users. Rapport helps users participate in meetings without leaving their offices by the provision of *virtual meeting rooms.* This perspective highlights important characteristics of face-to-face meetings that have been reproduced in the Rapport conference environment. Rapport also supports the conventional telephone service within its conferencing framework and allows application programs to execute within a conference. A common input/output environment which allows these facilities to be accessed by conference participants is provided.

Within Rapport, participants may start a conference by going into a meeting room and then inviting others to join them there. Similarly, when a person wishes to join a conference, he or she must enter the appropriate virtual meeting room. Once in a room, conference participants are able to hear each other and to talk at will. Rapport supports a variety of computers with a wide range of capabilities. As a result user views of a meeting room vary significantly.

Conferees using workstations running UNIX and the X window system are able to see a number of common displays, produced by application programs. Each shared application produces an identical display on each conferee's screen; these displays are analogous to blackboard or slide presentations in a face to face meeting. Conferees have a unique pointer which allows them to point to items of interest on each display window. The participants themselves are also represented in the Rapport display. This allows conferees to point to one another and for conferees to post signals, i.e. to "raise their hands" for attention. Rapport also provides each conferee with a *beeper*or message service. While a person is in a meeting, he or she will be notified of requests for new meetings. These request may be annotated with a description and appear asynchronously on the conferees display.

A single cluster prototype of Rapport has been implemented using a network of Sun workstations connected by both an Ethernet and a voice network. The custom voice network provides call control and conferencing facilities similar to those provided by existing PBX telephone networks and is accessible via headsets associated with each workstation. The current implementation of Rapport does not support video enhancements to support video are planned.

Rapport is typical of a number of recent systems which have as their aim the provision of the facilities found at face to face meetings within remote groups. The virtual room metaphor of Rapport has also been applied at Bellcore in the development of a system called Cruiser [Root 88]. Cruiser provides a virtual corridor of offices into which a user can look or enter. The system provides multi-media links to a number of offices within Bellcore which may be displayed when a user enters an office. The aim of cruiser is to mimic a more relaxed form of social interaction than that found in stricter conferencing systems. As technology improves it is envisaged that these systems will become more widespread and support more and more geographically dispersed groups. One example of this development is the MERMAID Conferencing system [Watabe 90] from NEC which provides distributed, real-time, multi-media conferencing using narrow-band ISDN.

5. Meeting Room Systems

The support of face to face cooperation represents the most recent and distinct research development in cooperative working. A typical approach to this form of computer support is to develop a meeting room furnished with a large screen video projector and a number of computer workstation/terminals, often these systems include a control terminal. A typical meeting room arrangement is shown in figure 10.

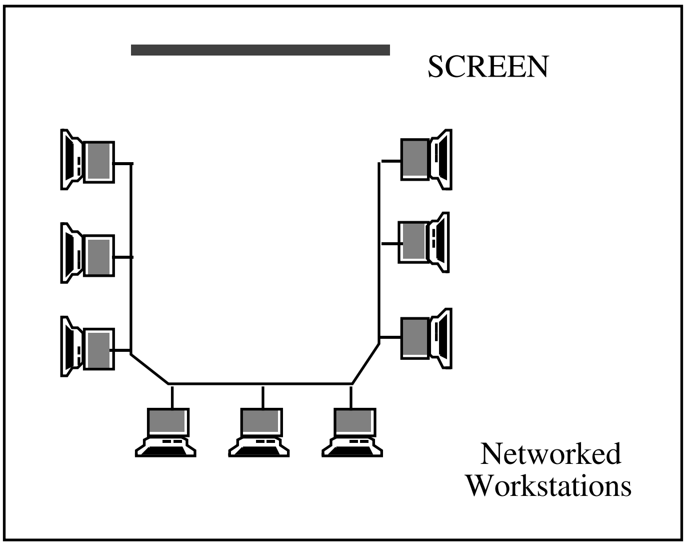

Figure 10 A typical Meeting room Arrangement

Most systems which employ meeting rooms to support cooperation among local groups belong to a class of systems know as *decision conferences*. Decision conferences are related to earlier Decision Support Systems [Ariav 85] but focus on improving decision making by groups rather than individuals. Decision conferences emphasis the use of structured decision processes, mainly involving statistical computer models but increasingly utilising models which embody collaborative notions such as the Delphi method.

A number of decision conferences have been developed both in industry and university research centres. In industry, systems include the GROUP DECISION AID of Perceptronics Inc. [Steeb 81] and the Decision Conference of Decision and Designs Inc. [Patterson 81]. In universities, decision conferences include the Planning Laboratory at the University of Arizona [Applegate 86] and the Decision Room [Gray 81]. A fundamental principle of the MIS laboratory like all decision laboratories is the emphasis on both statistical and analytic decision models.

A discussion of GDDS and Decision conferences is given in [Kraemer 88] who identifies a number of functions which Group Decision Systems can provided. Kraemer identifies the following software elements as being important to the construction of decision conferences:-

- Decision Analysis Software
- Modeling Software
- Voting Tally
- Display Software

In this section we will examine two example of meeting room systems, the CoLab system developed at Xerox Parc, and Project NICK [Cook 87] at the MCC. The CoLab project is interesting because it represents a move away from the reliance on quantitative models exhibited by other meeting room systems. Project NICK represents a significant development in meeting support systems because of its long term goal to develop models to formalise the meeting process and to build tools supporting these models.

5.1 CoLab

The collaborative laboratory (CoLab) at Xerox Parc focuses on computer support for co-located group work. CoLab does not incorporate formal decision models and quantitative techniques. Rather it supports writing and argumentation using verbal models and qualitative techniques through the manipulation of text and graphical images.

Physically, CoLab consists of bit mapped workstations communicating over a local area network (Ethernet) and a shared electronic *chalkboard*. The electronic chalkboard derives many properties from its non-electronic counterpart allowing the flexible placement of text and figures. In addition, many of the functions that are awkward or impossible with actual chalkboards are supported by this electronic equivalent. For example, text and images can easily be moved, and the information described on the chalkboard may be stored for use at a later date.

The developers of CoLab state that "a fundamental requirement for meeting tools is that they provide a co-ordinated interface for all participants". CoLab achieves this aim by the utilisation of a multi-user interface based on an abstraction they call *WYSIWIS*. In fact, CoLab was one of the forerunners in the development of the shared screen technology described in section 4.3. The term WYSIWIS(**W**hat **Y**ou **S**ee **I**s **W**hat **I** **S**ee) refers to the presentation of consistent images of shared information to all participants. A meeting tool can be described as *strictly WYSIWIS* if all meeting participants are presented with identical views including indication of where the others are pointing (this effect is often called *tele-pointing*).

In practice, strict WYSIWIS has a number of distinct problems. The display of cursors from multiple users is distracting, while the small granularity transmission of data required is computationally expensive. To combat these problems CoLab uses a relaxed version of WYSIWIS [Stefik 87b]. For example, Stefik considers it useful to differentiate between public interactive windows accessible to the entire group and private windows with limited access.

In addition to WYSIWIS, a key issue in meeting tool design is supporting parallel activities. For parallel actions, a task must be broken up into appropriately sized operations that can be performed independently by different group members. In addition to allowing parallel activities meeting tools must be aware of conflict and provide some form of conflict resolution between parallel actions.

5.2. The Project NICK Model

The software technology program of the MCC is investigating the early stages of the design process, before requirements are finalised, for large-scale distributed systems. Much of the work of requirements specification, design, and implementation involves meetings. Face to face meetings are an important activity during this phase of a project and, as a result, the MCC began an investigation into the analysis and augmentation of meetings.

Project NICK [Cook 87] focuses upon a small class of meetings restricting itself to 'small' design meetings which they describe as less than fifty members. They are concerned only with those meetings whose activities include brain-storming, exploring, defining design structure, analysing issues, making task assignments, and resolving problems. Meeting research within the project is based on a two-pronged approach, *building theories* and *building systems.*

The building theories aspect of the work centres around the careful definition of basic notions and the generation of hypotheses. These hypotheses are implemented within systems which are then analysed generating new hypotheses. Subsequently effort within theory building has focused on meeting analysis. The building systems approach of project NICK includes the construction of electronic meeting aids. The project’s meeting aid environment includes

- An Electronic Blackboard
- Networked Personal Computers
- Software for Recording Meeting Notes

Generally information generated at a meeting is captured, analysed, and made available for use or presentation at subsequent meetings. These three phases constitute the basic elements of the project NICK model (figure 11)

- A capture component
- An analysis component
- A presentation component

Figure 11 The Aspects of Project NICK

A typical meeting (for example, meeting X) has inputs from many sources and outputs a variety of information. This information is held within information repositories each of which *capture* a subset of the meeting information. The model distinguishes between repositories that have intelligent, sophisticated interpretation and selection mechanisms(called *pots*) and those that do not.

The information contained within a meeting's set of repositories is input into the *analysis* component of the model. This component synthesises and summarises the information from these repositories. It includes information analysis to capture, understand and improve the syntax and semantics of meetings and useful transformations of input information for enhanced presentation. Such as, the transformation of typed input to spoken output.

Part of the on-going work within the analysis phase of the project NICK model is the development of the notion of *information frames* as a presentation technique. The concept of frames separates information content from presentation attributes and allows analysis and correlation of the information.

The final phase of the model is *presentation* of the analysed, transformed information for subsequent meetings. The presentation component of the NICK model manages the display of computerised information and provides editing facilities for brain-storming and creative ad-hoc talks.

As part of their model project NICK catagorises the types of information that may be presented, manipulated and captured during a meeting according to the amount of its *structure* (binary, structure or unstructured), and according to the degree of its *sharing* (private, subgroup, public) . This categorisation is depicted by figure 12.

	Private	**Subgroup**	**Public**
Binary	Checkoff List	Support Flags	Voting
Structured	Lists	Facilitation Messages	Presentations Lists
Unstructured	Bitmap	Notes and Comments	Electronic Sketch Board

Figure 12 Meeting Information Matrix

Project NICK is concerned with the development of a three phase model for meetings and the subsequent support of this model through pre-meeting, during- meeting and post-meeting aids. Pre-meeting aids include the development of software which supports the design of sophisticated diagrams and slides. Analysis software is being developed as an aid to be used during a meeting to allow statistics such as time spent on an agenda topic to be identified and reasoned about. Current research is concerned with the development of post-meeting software tools and analysis formalisms.

6. Co-Authoring and Argumentation Systems

A significant number of documents result from the work of more than than one author. The aim of co-authoring systems is to support the cooperation necessary between these co-authors in document production.

The general model adopted by these systems is that of asynchronous co-operation with each user working independently on a portion of the document. Reviews and comments are added to the document by annotating sections of the document. However while the mode of interaction in co-authoring systems is asynchronous co-authoring systems are not distinguished by the location of users and some co-authoring and outline processor and argumentation systems are used when the participants are co-located.

Because of their model of cooperation and annotation co-authoring systems often use hypertext technology. The term *hypertext* describes any system employing non linear structuring of text, graphics, and other media. Hypertext systems normally form linked network structures with data (usually text or graphics) in the nodes and occasionally typing information on the links (see [Conklin 87a] for an overview of hypertext systems). Hypertext documents resemble nets of connected nodes with each link between nodes denoting an association between the information held in the nodes (figure 13).

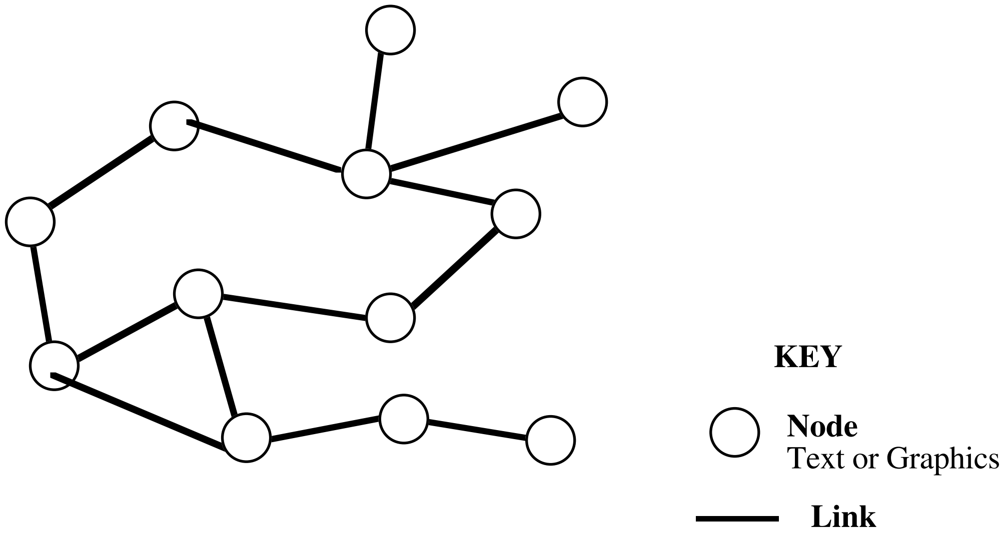

Figure 13 A typical hypertext network

Hypertext systems are successful in structuring information in such a manner that the user's ability to process it is enhanced [Marchionini 88]. A general view of argumentation and cooperative authoring systems is as multi-user hypertext systems, where the hypertext document (or network) is constructed by a number of users adding nodes to the network in an independent manner. However, the provision of tools which deal explicitly with the interaction arising in collaborative hypertext settings are relatively recent.

An example of the the general approach being adopted is the work currently being undertaken at Tektronix's Research Laboratory on an extension to Neptune [Delise 86] called *contexts*.

In existing Hypertext systems all users make changes within a shared storage system. This situation results in numerous conflicts when several authors are collaborating on the production of a document. Current hypertext systems provide inadequate support for the resolution of these conflicts. Delise enumerates these deficiencies:-

- Limited Organisation: a collaborative environment should provide a means of organising related sets of hypertext nodes and links. For example, a set may contain all the nodes and links that comprise a specific document.
- Lack of partitioning: there should some mechanism which allows teams of authors to work together in independent hypertext partitions without risk of interference, and then to allow those independent partitions to be joined at carefully controlled intervals.
- Little Version and Configuration Control: users should have the ability to build version trees and subsequently install specific branches as the primary version. In addition, some form of configuration management support allowing configurations to be built and used should be provided.
- Lack of distribution: the hypertext database should be distributed to support large teams collaborating on a common project.

Contexts attempts to provide a mechanism which is powerful enough to support a variety of users, resolving the above problems, while remaining conceptually simple. Contexts is based on the principle that users have a private view of a hypertext graph and may make modifications within this view. When alterations are completed, they can be released to other project team members by merging the users private view (*context*) with the shared *master* view.

When a context is created it is empty. A collection of nodes and links can be moved into the new context from any other existing context using a *merge* operation. A context is often used as a private workspace, where the intention is to make modifications to nodes and links that already exist in some other master context. The new context will contain new *instances* of existing nodes and links. The notion of multiple instances of a particular entity are used to facilitate the merging of different contexts and the detection of update conflicts across contexts.

Other systems following this line of development, each with their own individual peculiarities include, Intermedia [Garret 86], Xanadu [Nelson 80] and Notecards [Trigg 86].

This section examines two classes of system which represent applications of multi-user hypertext technology. Section 6.1 examines argumentation systems by considering gIBIS while section 6.2 examines co-authoring systems by considering Quilt as an exemplar.

6.1 Argumentation Systems

Argumentation systems support the structured development of multi-party arguments and negotiation. The gIBIS [Conklin 87b] system supports the argumentation process undertaken by system designers. The system is based on a well known and simple model of design deliberation called Issue Based Information System, or IBIS [Rittel 73].

The IBIS method is based on the principle that the design process is a conversation between a number of protagonists each holding a specific viewpoint. The IBIS model focuses on the Key *Issues* in the Design problem. Each Issue can have many *Positions*, where a Position is a statement or assertion which resolves the Issue. Each of an Issues

Positions, in turn, may have one or more *Arguments* which either support that position or object to it. Thus each separate Issue is the root of a tree, with the children of the Issue being Positions and the children of the Positions being Arguments. The application of the IBIS model within gIBIS allows a small set of legal rhetorical moves to be constructed, the legal gIBIS moves are shown below in figure 14

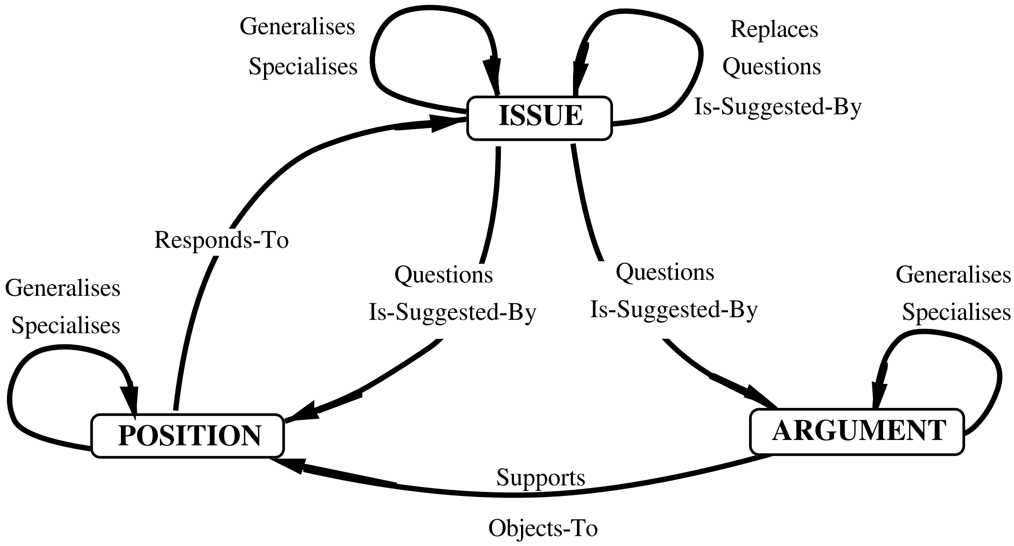

Figure 14 : The set of legal rhetorical moves in gIBIS

The gIBIS tool supports simultaneous access and update of issue groups by multiple users on a common local area network. It provides the necessary concurrency control, locking, and update notification to allow real time interactive network construction by teams of cooperating designers.

gIBIS represents a particular solution which supports a specific approach to collaboration. As such the tool is limited in that it supports only a single perception of collaboration and argumentation within the design process. However, the method and the tool has proven to be successful [Yakemovic 90] with its underlying model easily applicable to design process.

6.2 Co-Authoring Systems

Co-authoring systems are a class of cooperative systems which apply the principles of hypertext technology in a cooperative setting. Like argumentation systems co-authoring systems exploit an asynchronous model of cooperation irrespective of the physical proximity of the users. The Quilt system [Fish 88] developed at Bell Communications is representative of the general principles used by most co-authoring systems.

Quilt is aimed at supporting the work of existing groups of authors who have an established pattern of cooperation. The premise is that Quilt is used after some form of planning/ brain-storming meeting which establishes the nature of the cooperation between the authors.

Quilt is designed to be editor independent and provides a framework for the sharing of comments and annotations. A document in Quilt consists of a base and nodes linked to the base using hypertext techniques. The aim is that these nodes act in a similar way to paper notes, Post-its, and margin comments in paper documents.

The general principle of cooperation in Quilt as in most co-authoring systems is that the users read through a publicly available document annotating the document to reflect their comments. To achieve this the designers of Quilt believe that a general hypertext graph is unnecessary and a tree structure is sufficient for supporting most cooperative authoring.This restriction on the more general hypertext linking techniques imposes a particular style of cooperation and interaction.

At any time a Quilt comment tree will consist of :

- A current base document consisting of the text and other materials the authors consider publicly available
- Revision suggestions, which are available in a form where users with appropriate permissions can swap existing document paragraphs with them.
- Comments with an associated set of permissions which in turn distinguishes them to be one of three different types of comment.
    1) *Private Comment* visible only to the creator.
    2) *Public comment* can be read by anyone with permission to read the parent
    3) *Directed message* whose existence is displayed only to named individuals or groups. Directed messages may make use of a notation facility to inform users of their existence.

Quilt provides facilities for making the document comment tree more active by the attaching triggers which can cause a range of possible commands to be executed. A possible use of triggers is maintaining deadlines for document production. Quilt also provides facilities for the tracking of user actions and a message system to support their coordination.

The principles employed in Quilt as a co-authoring systems are mirrored by those found in other co-authoring systems including the Co-Author system [Hahn 91] at Passau, and the GROVE system [Ellis 88] at MCC each of which have their own peculiarities and make varying use of hypertext mechanisms. The premise is that each item within a document can by one of a number of types reflecting the kind of comment being made. The process of generating comments from an original document consists of a number of authors and reviewers constructing a structured document for these typed nodes reflecting their comments

8. Summary

In this paper cooperative systems are considered as belonging to one of the classes introduced in section 2.1. These classes are message systems, conferencing systems, co-authoring systems and meeting support systems. While cooperative systems can be considered as belonging exclusively to one of the four classes identified it has to be remembered that people cooperate in a variety of ways and each class of systems has to coexist to fully support cooperation. In fact, a number of cooperative systems combine a range of functionality from each of the classes identified. This is most notable in commercially available computer conferencing systems which include messaging facilities as an integral part of their system.

Message systems are typified by their development from wide area store and forward networks. As a result , message systems tend to be used to exchange message objects between remote groups in an asynchronous manner. In contrast computer conferencing systems, which began in a similar way to message systems, have adopted technological advances in networking services by developing a number of variations of conference systems. Most notable of these has been the emergence of real-time conferencing systems and the subsequent development of multi-media conferencing systems.

In contrast to their remote counterparts, meeting support systems have developed by providing support for meeting activities within a single conference room. This has led to the development of a number of purpose built conference rooms which contain the machines and appropriate software tools for various classes of meeting. A number of the principles discovered in these meeting rooms have been more generally adopted by cooperative systems. The most notable example of this effect has been the emergence of shared screen facilities with associated WYSIWIS techniques.

Cooperative authoring and argumentation systems were considered as the final class of systems. Co-authoring systems aim to support some of the most fundamental parts of cooperation by supporting the negotiation processes. Rather than directly modelling and supporting the process of negotiation these systems provide a framework for negotiation to take place in. Most of the systems developed to date have relied on hypertext technology to provide this framework.

When classifying cooperative systems two classification mechanisms were adopted; the location of the users, and the mode of cooperation. It was argued that the modes of cooperation were either synchronous or asynchronous. Synchronous cooperation being characterised by the simultaneous presence of all cooperating users. Synchronous systems included meeting support systems and real-time conferencing systems while asynchronous systems included message handling and co-authoring and argumentation systems.

While the division of cooperation into either synchronous or asynchronous is useful it still reflects a primarily technological perspective on cooperative systems reflecting the distinction between store and forward and real-time communication systems. An alternative classification may be derivable from a closer consideration of the cooperation being undertaken. People cooperate on a range of problems each of which has a different amount of structure. Similarly, the cooperation involved has different levels of structure, ranging from the very structured co-operation of systems based on speech act theory to the very unstructured cooperation of the CoLab systems.

Future cooperative systems will require research from a range of disciplines including sociologists, cognitive scientists and computer scientists. The success of future cooperative systems will depend on the development of appropriate design and evaluation techniques. These techniques are currently in a very early stage and will shape the development of CSCW as a research discipline.

CSCW will also affect computer technology and will impact in the future research challenges of a number of areas of computer science. Most notably CSCW presents a new set of challenges to communication technology both in the development of new high-speed network services and the development and adoption of standards. The development of mechanisms which allow inter-operability between CSCW applications and the environmental support for these applications will become increasingly important.

The wide spread acceptance of multi-media technology and the development of multi-media communication and workstations will be important in the future development of CSCW. Similarly CSCW is likely to have a significant impact on future user interface and window management systems particularly with the adoption of shared screen technology. Finally, the concept of information sharing and the mechanisms used to store shared information will be directly impacted by the requirements of CSCW systems.

Appendix A Freedom and Control in Cooperative Systems

People work together to solve a wide variety of problems using different forms of cooperation for each class of problem. Cooperative problems can be though of as existing at some point on a spectrum ranging from poorly defined unstructured problems at one end to highly structured prescriptive tasks at the other. Unstructured problems are those requiring creative input from a number of users to solve them in some manner which cannot be detailed or described in advance; design is a good example of such an activity. Prescriptive tasks, on the other hand, represent the routine procedural cooperative mechanisms used to solve problems which have existing group solutions, for example the invoicing procedures used in large organisations. Prescriptive tasks respond well to detailed control of cooperation while undefined problems require a significant degree of freedom to be exercised by the cooperative system.

The control provided by cooperative systems is an additional means of classification which highlights the level of automation each cooperative system provides. The degree of freedom allowed by each type of system provides depth to the classification shown in section2. A significant area of research in cooperative systems hinges on the amount and form of control cooperative systems provide. If the control is too strong and restrictive then people lose the freedom needed to express the cooperation they require.

Two general modes of cooperation identified in section 2; synchronous and asynchronous interaction. The problem of describing and controling cooperation in asynchronous systems is the most mature of the two and accordingly most work has been done in this area. The problem is currently being addressed by two distinct research areas; speech act systems and office procedure systems.

A.1 Speech Act Systems

Much research effort within cooperative work has attempted to apply a linguistic approach to computer supported cooperation based on speech act theory. This theory has developed from the linguistic work of Austin [Austin 62] and considers language as a series of actions. For example statements such as "I pronounce you man and wife are" are actions. Searle [Searle 75] identifies five fundamental methods of interpreting such a speech act :-

- Assertive, commit the speaker to something being the case.
- Directive, attempt to get the hearer to do something.
- Commissive, commit the speaker to some future course of action.
- Declaration, bring about the correspondence between the propositional content of the speech act (e.g., pronounce a couple married).
- Expressive, express a psychological state about a state of affairs.

Speech acts are elements within larger conversational structures [Flores 81] which define the possible courses of action within a conversation between two actors. One class of conversational structure of direct relevance to cooperation are what Winograd and Flores [Winograd 86] term *conversations for action.* In such conversations an interplay of *Directives* and *Commissives* are directed towards explicit cooperative action. Winograd informally calls directives and commissives *requests* and *promises* respectively.

In a conversation for action, one party (A) makes a request to another (B). A's request has certain conditions of satisfaction which describe a future course of actions for B. After the initial request B can:-

- Accept: (and commit himself to satisfying the conditions);
- Decline: and end the conversation;
- Counter offer : with alternative conditions

Each of these in turn has its possible continuations.

Winograd and Flores use graphs, similar to state transition diagrams, to plot the basic course of such a conversation (see figure A1).

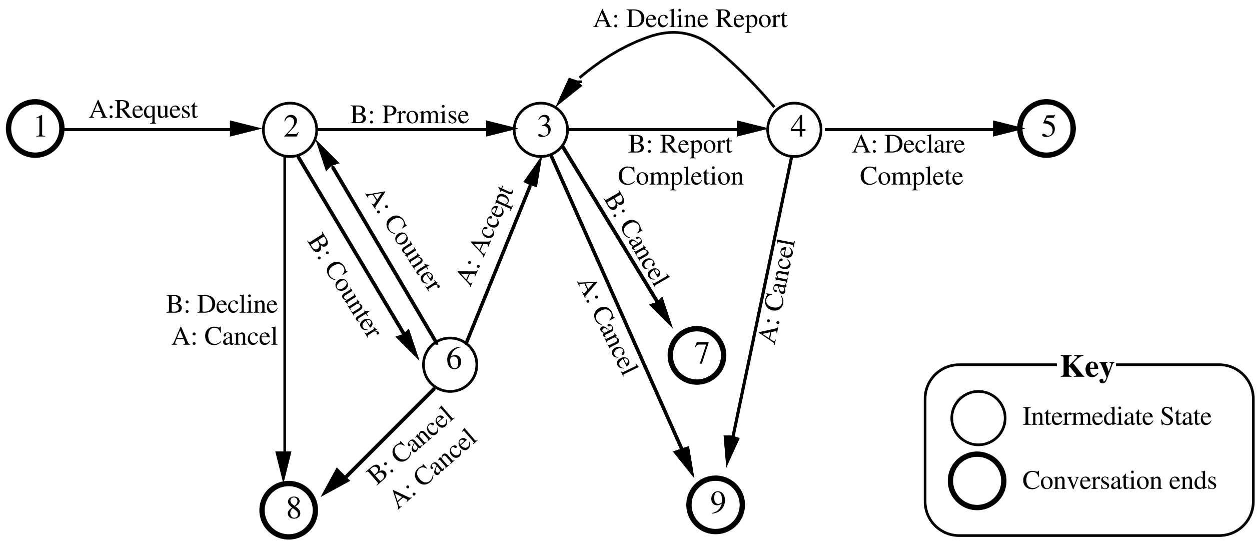

Figure A1 A conversation for Action

Conversations for action represent only one possible form of conversation. Winograd distinguishes several additional kinds of conversation which accompany conversations for action, including; *conversations for clarification, conversations for possibilities,* and *conversations for orientation.*

Speech Act theory as applied by Winograd and Flores forms the basis for several computer systems and represents a fundamental area of research within cooperative working. Systems which make direct use of speech act theory include:-

- The Coordinator system [Winograd 87] developed by Winograd and Flores.
- MONSTR, a system for structuring a software maintenance environment [Cashman 80]
- XCP - a co-ordination system for supporting office procedures [Sluzier 84]
- The CHAOS project [De Cindio 86] which is concerned with the use of speech act theory to support co-ordination within an office environment.

Three points of note about the work of speech act theory are:-

- The complexity of the nets used to describe even the simplest conversation is high (see figure A1).
- The restrictive nature of conversation structures which allow only a small range of possible messages at any particular time.
- The majority of work within co-ordination systems centred on speech act theory is based around one-to-one conversation between a requester role and a receiver role rather than more general patterns of group interaction.

A.2 Office Procedure Systems

As Grief observes [Grief 88], the research fields of Office Information Systems and CSCW are similar. Both are interdisciplinary and strongly concerned with social and technical issues. As a result, office information systems have been a successful source of application for CSCW. However, as [Wilson 88] concludes, the terminology and approach used within office automation differs from that of the CSCW community. In the office automation community, the notion of group communication (as used within the messaging and CSCW communities) is replaced by the notion of *Office Procedures.*

Office procedures describe tasks performed within an office in terms of the combined effect of a number of small sub-tasks or procedures. Examples of procedures range from the well defined and understood (invoicing, for example) to less well defined tasks such as the generation and approval of an office budget. The cooperative research within office automation has concentrated on finding a unifying language which allows the specification of office procedures and a description of their interaction.

Within their paper on the inter-relation of group communication and office automation, Speth and Prinz [Prinz 87] identify three main models based on how the procedural rules are held. An equivalent classification is given in [Bracchi 84]:-

- Data Based Models where co-ordination knowledge is stored centrally and often routed by means of forms. For example, OBE [Zloof 82] and Officetalk -Zero [Ellis 80]
- Process- Based Models concentrate on the representation of concurrency as a means of describing office systems. Process-based systems include systems such as POISE [Croft 84].
- Agent Based Models apply AI techniques such as goal based planners and active agent models. Office tasks are modelled using known AI modeling techniques or systems and an inference engine is used to generate and execute task plans. Systems within this category include AMS [Tueni 88] and POLYMER [Croft 88].

Office automation procedure specification relies on some language to specify the office tasks, which is then used to coordinate cooperation within the office. Object-oriented languages have proven particularly amenable for this task forming the basis of the early Officetalk-D system and the OTM [Lochosvsky 88] system.

Many of the systems which used procedural languages to describe office procedures found them unsatisfactory. As a result of this dissatisfaction, office automation systems have started to utilise AI technology. The POLYMER system, for example, has a task manager which uses an AI based planner to execute actions and coordinate various interaction via an *execution monitor* in order to achieve the goals of the plan (figure A2).

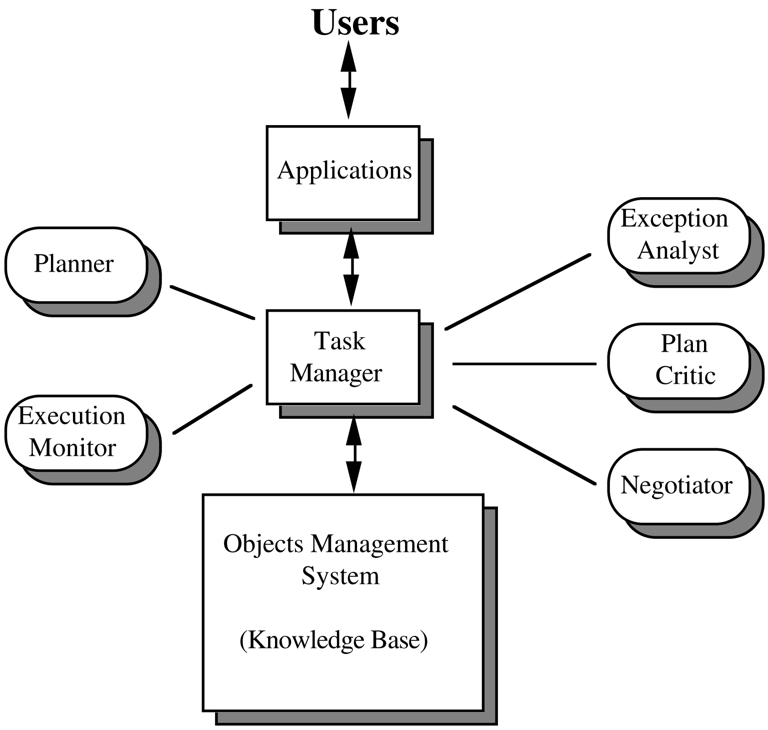

Figure A2 The Polymer Architecture

Plans within POLYMER are expressed in a script like formalism called GRAPPLE [Broverman 86]. The task manager within POLYMER is able to handle exceptions to these plans. When an exception arises the *plan critic* determines its effect on the plan. The *exception analyst* attempts to find an explanation for the exception and the *negotiator* uses the information from the analyser to determine through interaction with the user and autonomous agents how best to resolve the exception.

Acknowledgements

Thanks are due to Professor Ian Sommerville and Dr John Mariani for their helpful comments on earlier drafts of this paper.

References

[Ackerman 90] Ackerman M.S., Malone T.W., ‘ 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.

[Ahuja 88] Ahuja S.R.,Ensor J.R.,Horn D.N.' The Rapport Mulimedia Conferencing system', in Allen R.B.(ed) COIS88 Proceeding conference on Office Information Systems, March 23-25,1988, Palo Alto, California,.

[Applegate 86] Applegate L., Knonsynski, Nunamaker J.F. ' A Group Decision Support System for Idea Generation and Issue analysis in Organisational Planning', in proceedings of CSCW 86, Austin , Texas, December 1986.

[Ariav 85] Ariav G., Ginzberg ' DSS Design: Asystemic View of Decision Support', Comm ACM, Vol 28, No 10, October 1985.

[Austin 62] Austin J.L. 'How to do things with words', Harvard University Press, 1962.

[Bannon 91] Bannon L., Schmidt K.’ CSCW: Four Characters in Search on context’ in J.M. Bowers and S.D. Benford (eds): Studies in Computer Supported Cooperative Work. Theory, Practice and Design, North-Holland, Amsterdam, 1991.

[Bonfiglio 91] Bonfiglio A, Malatesta G., Tisato F. ‘Conference Toolkit : A frame work for real time conferencing’ in J.M. Bowers and S.D. Benford (eds): Studies in Computer Supported Cooperative Work. Theory, Practice and Design, North-Holland, Amsterdam, 1991.

[Borenstein 88] Borenstein N.F., Thyberg C.A. ‘ Cooperative work in the Andrew Message Systems’ , in proceedings of CSCW88, Portland, Oregon, September 1988

[Bowers 88] Bowers J., Churcher J., Roberts T. 'Structuring Computer Mediated Communication in COSMOS',in R. Speth(ed) EUTECO 88: Research into Networks and Distributed Applications, Vienna, Austria, April 20-22,1988.

[Bracchi 84] Bracchi G., Pernici L.' The design requirements of Office Systems' ACM trans. on Office Information Systems, Vol 2 No 2, April 1984.

[Brobst 86] Brobst S. A., MaloneT., *et al*. 'Toward intelligent message routing systems' In:, R Uhlig (Ed.), Computer Message systems -85. Proc. of the 2nd International Symposium on Computer Message Systems. North Holland, Amsterdam, 1986.

[Broverman 86] Broverman C, Huff K, Lesser V. ' The role of plan recognition in the design of an intelligent user interface', Proceedings of the IEEE Conference on Man,Machine and Cybernetics, pp836-868, 1986

[Cashman 80] Cashman P. and Holt A.W., ' A communication Oriented approach to structuring the software maintenance environment', ACM SIGSOFT, Software Engineering notes, Vol 5, No 1, January 1980.

[CCITT 87] CCITT (1987) ' Draft Recommendation on Message Handling Systems, X 400, Version 5' , November 1987.

[Churcher 88] Churcher J and Bowers J ' A structure Definition language for COSMOS:a prelimary report', COSMOS Document 43.1.

[Conklin 87a] Conklin J. ' Hypertext: An Introduction and Survey', IEEE Computer September 1987, pp 17-41.

[Conklin 87b] Conklin J. ' gIBIS: A Hypertext Tool for Team Design Deliberation',Proceeding of Hypertext 87, November 1987,pp 247-251.

[Cook 87] Cook P., Ellis C., Graf M., *et al* ' Project NICK: Meetings Augmentation and Analysis', ACM trans. on Ofiice Information Systems, Vol 5, No 2, Aprill 1987,pp 132-147.

[Croft 84] Croft W.B, Lefkowitz 'Task support in an Office System', ACM trans. on Office Information Systems, Vol 2 No 3, July 1984.

[Croft 88] Croft W.B, Lefkowitz L.S. ' Using a Planner to support office work' in Allen R.B.(ed) COIS88 Proceeding conference on Office Information Systems, March 23-25,1988, Palo Alto, California.

[Crowley 90] Crowley T., Milazzo P., Baker E., Forsdick H., Tomlinson R. ' MMConf: An infrastructure for building shared multimedia applications', in proceedings of CSCW 90, Los Angeles, CA, October 7-10 1990, ACM press , ISBN 0-89791-402-3.

[Danielson 86] Danielson T., Panoke-Babatz U. *et al* ' The AMIGO project: Advanced Group Communication Model for Computer-based Communication Environment', in proceedings of CSCW 86,Austin,Texas,December 1986.

[De Cindio 86] De Cindio F., De Michelis G.,*et al '* CHAOS as a Coordinating Technology', in proceedings of CSCW 86, Austin, Texas, December 1986.

[Delise 86] Delise N.M., Schwartz M.D. 'Neptune: A Hypertext system for CAD applications', In proceedings of the 1986 ACM-SIGMOD International Conference on the Management of Data, Washington, D.C., May 28-30, ACM, New York,1986, pp 132-143.

[Ellis 80] Ellis C A., Nutt G 'Office Information Systems and Computer Science', ACM Computing Surveys, March 1980.

[Ellis 88] Ellis, C., Gibbs, S.J. and Rein, G., "Design and Use of a Group Editor ", MCC Technical Report Number STP-263-88, Sept. 1988.

[Fanning 86] Fanning T., Raphael B 'Computer Teleconferencing: Experience at Hewlett Packard', in proceedings of CSCW 86,Austin,Texas,December 1986.

[Fish 88] Fish R.S., Kraut R.E., Leland M.D., Cohen M. ' Quilt: A colloborative tool for cooperative writing', in Allen R.B.(ed) COIS88 Proceeding conference on Office Information Systems, March23-25,1988, Palo Alto, California,.

[Flores 81] Flores C.F. and Ludlow J 'Doing and speaking in the office':, in G Fick and R. Sprague (Eds), DSS:Issues and Challenges,London, Pergamon Press,1981.

[Forsdick 82] Forsdick H.C., Thomas R.H. 'The Design of Diamond: A distributed Multimedia Document System', Technical Report 5402, Bolt Beranek and Newman, Inc., October 1982.

[Garret 86] Garret L.N., Smith K. and Meyrowitz ' Intermedia: Issues, Strategies and Tactics in the Design of a Hypermedia Document System', in proceedings of CSCW 86, Austin, Texas,December 1986.

[Gray 81] Gray P. *et al* ' The SMU decision room project', In proceedings of the first International Conference on Decision Support Systems',Execucom Systems Corporation, Austin, Texas, pp122-129.

[Grief 88] Grief I.(Ed) 'Computer Supported Cooperative Work: A Book of Readings', Morgan Kaufmann Publishers, San Mateo CA., 1988, ISBN 0-934613-57-5.

[Gust 88] Gust p ' Shared X: X in a distributed group work environment', presented at the 2nd Annual X conference, M.I.T, Boston, Jnauary 1988.

[Hahn 91] Hahn U., Jarke M., Kreplin K.et al. ‘ 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.

[Hiltz 78] Hiltz S.R and Turoff M. 'TheNetwork Nation: Human Communication via Computer', Addison Wesley, New York,1978, ISBN 0-201-03141-8.

[Hiltz 85] Hiltz S. R , Turoff M. 'Structuring computer-mediated communication systems to avoid information overload', Comm ACM Vol 28 No 7 July 1985.

[Kraemer 88] Kraemer K L, Kling J L ' Computer based systems for cooperative work and group decision making', ACM Computing Surveys, Vol 20, No 2, June 1988.

[Kreifelts 86] Kreifelts T. Woetzel ‘ Distribution and error handling in an office procedure system’ Proceedings IFIP WG 8.4 Working Conference on Office Systems Methods and Tools, Pisa, Italy, 1986, p 197-208.

[Lantz 86] Lantz K.A. 'An experiment in integrated multimedia conferencing', in proceedings of CSCW 86,Austin,Texas,December 1986.

[Lauwers 90] Lauwers J.C., Lantz K.A. ‘ Collaboration awareness in support of collaboration transparaency: Requirements for the next generation of shared window systems’ Proceedings of CHI ‘90 Seattle, Washington April1-5, 1990, ACM press, ISBN-0-201-50932-6.

[Lee 90] Lee J ' SIBYL: A tool for managing group decision rationale', in proceedings of CSCW 90, Los Angeles, CA, October 7-10 1990, ACM press , ISBN 0-89791-402-3.

[Lochosvsky 88] Lochosvsky F.M, Hogg J.S., Weiser S.P., Mendelson A. O. 'OTM: Specifying Office Tasks', ACM SIGOIS Conference on Office Information Systems, March 23-25, Palo Alto CA 1988.

[Malone 87] Malone T.W., Grant K.R., Turbak F.A., Brobst S.A., Cohen M.D. 'Intelligent Information Sharing Systems'. Comm. ACM Vol. 30, no. 5, 1987.

[Malone 88] Malone T W, Lai K, ' Object Lens: A Spreadsheet for Cooperative Work', in proceedings of CSCW88, Portland, Oregon, September 1988.

[Marchionini 88] Marchionini G., Schneiderman B. ' Finding Facts vs Browsing Knowledge in Hypertext Systems', IEEE Computer, Jan 1988, pp 70 - 79.

[Mortensen 85] Mortensen E.'Trends in electronic Mail and it's role in office automation' Electronic Publishing Review Vol 5 No 4 Dec 1985 pp257-68

[Nelson 80] Nelson T.H. ' Replacing the Printed Word: A complete literary system ' IFIP Proc., October 1980, pp 1013-1023.

[Palme 84] Palme J. 'Survey of Computer Based Message Systems' In B.Shackel (Ed),INTERACT-84.

[Patterson 81]	PattersonJ.F., Randal L.S., Steward R. ' Advisory Decision Aids: a prototype', Technical report PR 80-27-312. McLean, VA: Decision and Designs, Inc.
[Poggio 85]	Poggio A., *et al*: 'CCWS: A Computer Based Multi-media Information System' IEEE Computer, Vol18,No10,October 1985.
[Prinz 87]	Prinz W., Speth R. ' Group Communication and related Aspects in Office Automation' in Proceedings of the IFIP TC 6/WG 6.5 Working Conference on Message Handling Systems, Munich, 27-29 April 1987.
[Pullinger 86]	Pullinger D.J. ' Chit-Chat to Electronic Journals: Computer Conferencing Supports Scientific Communication', IEEE trans. on Professional Communications, Vol PC 29, No 1, pp23-29, March 1986
[Rittel 73]	Rittel H., Webber M. ' Dilemas in a General Theory of Planning', Policy Sciences, Vol 4. 1973.
[Rodden 91]	Rodden T., Blair G., ‘ CSCW and Distributed Systems: The Problem of Control’ in proceedings of ECSCW’91,Amsterdam, 25-27th September 1991.
[Root 88]	Root R. W., ' Design of a Multi-Media Vehicle for Social Browsing', in proceedings of CSCW88, Portland, Oregon, September 1988.
[Sarin 85]	Sarin S., Grief I.: 'Computer-Based Real time Conferencing Systems', IEEE Computer October 1985, pp 33- 45.
[Searle 75]	Searle J.R. 'A Taxonomy of illocutionary acts'', in K.Gunderson (Ed.), Language,Mind and Knowledge, Minneapolis, University of Minnesota, 1975.
[Sheperd 90]	Sheperd A, Mayer N., Kuchinsky A., ‘ Strudel- An extensible elctronic conversation toolkit’, in proceedings of CSCW 90, Los Angeles, CA, October 7-10 1990, ACM press , ISBN 0-89791-402-3.
[Sluzier 84]	Sluzier S and Cashman P.M 'XCP: An experimental tool for Supporting Office Procedures', in IEE 1984 proceeding of the first international conference on office automation.
[Steeb 81]	Steeb R., Johnston S.C ' Automated management of communications with remote systems: a decision analysis approach'
[Stefik 87a]	Stefik M., Foster G., *et al* 'Beyond the chalkboard: computer support for collaboration and problem solving in Meetings', Comm ACM Vol 30, No 1, January 1987.
[Stefik 87b]	Stefik M., Bobrow D.G., *et al* 'WYSIWIS Revised: Early experiences with Multiuser Interfaces', ACM trans. on Ofiice Information Systems, Vol 5, No 2, Aprill 1987, pp 147-168.
[Trigg 86]	Trigg R., Suchman L., Halasz F. 'Supporting Collaboration in Notecards', in proceedings of CSCW86, Austin, Texas, December 1986.

[Tueni 88] Tueni M., Li J., Fares P. ' AMS: A Knowledge Based Approach to Task representation , Organisation and Coordination',in Allen R.B.(ed) COIS88 Proceeding conference on Office Information Systems, March23-25,1988, Palo Alto, California.

[Watabe 90] Watabe K., Sakata S, Maeno K et al ‘ Distributed Multiparty Desktop Conferencing System: MERMAID’, in proceedings of CSCW 90, Los Angeles, CA, October 7-10 1990, ACM press , ISBN 0-89791-402-3.

[Wilbur 88] Wilbur S.B., Young R.E. 'The COSMOS Project : A Multi-Disciplinary Approach to Design fo Computer Supported Group Working', in R. Speth(ed) EUTECO 88: Research into Networks and Distributed Applications, Vienna, Austria, April 20-22,1988.

[Wilson 88] Wilson P ' Key research in Computer-Supported Cooperative Work' , in R. Speth(ed) EUTECO 88: Research into Networks and Distributed Applications, Vienna, Austria, April 20-22,1988.

[Winograd 86] Winograd T. and Flores F. 'Understanding Computers and Cognition A new foundation for desing', Ablex Publishing Corporation, Norwood,New Jersey,1986,ISBN 0-89391-050-3

[Winograd 87] Winograd T. 'A language/action perspective on the design of cooperative work', Stanford University Department of Computer Science Technical Report , STAN-CS-87-1158.

[Yakemovic 90] Yakemovic K.C., Conklin J., ' Report on a Development Project Use of an Issue Based Information System', in proceedings of CSCW 90, Los Angeles, CA, October 7-10 1990, ACM press , ISBN 0-89791-402-3.

[Zloof 82] Zloof M : ' Office by Example: a business language that unifies data and word processing and electronic mail', IBM systems journal Vol 21 No 3, 1982.

[Ellis, 1991 #376]