The Challenges of CSCW for Open Distributed Processing

Gordon S. Blair and Tom Rodden

Department of Computing, Lancaster University, Bailrigg, Lancaster LA1 4YR, U K.

Abstract

The user-centred philosophy of Computer Supported Cooperative Work (CSCW) challenges the established principles of many existing technologies. In turn, the development of CSCW is dependent on the facilities offered by these technologies. It is therefore important to examine and understand this relationship. This talk will review key developments in CSCW and consider the potential impact of these developments on ODP. The talk will also highlight a number of outstanding issues raised by CSCW. It is hoped that this talk can stimulate closer cooperation between the two communities in developing future IT standards and services.

Keyword Codes: C.2.4; H.5.1; K.4.3
Keywords: Distributed Systems; Multimedia Information Systems; Organizational Impacts

1. INTRODUCTION

Computer Supported Cooperative Work (CSCW) has emerged as an identifiable research area which focuses on the role of the computer in group work. The majority of CSCW applications are fundamentally *distributed* and are dependent on the facilities provided by existing distributed systems platforms. They also typically require *open* solutions to distributed processing. For example, many CSCW applications support co-operative work between different departments, sections or even organisations with each likely to have different computer equipment, operating systems, work practices and management policies. The emerging standards for Open Distributed Processing (ODP) are therefore of considerable relevance to the CSCW community. We argue that CSCW is also of considerable relevance to ODP:

i) CSCW is generating a wide range of distributed applications including co-authoring systems, desktop conferencing systems and decision support systems,

ii) many CSCW applications have stringent requirements for the underlying infrastructure, e.g. in terms of multimedia communication or real-time interaction,

iii) CSCW and ODP share a holistic view of systems design, ODP with its concept of viewpoints and CSCW with its inter-disciplinary approach.

This paper examines the likely impact CSCW systems will have on ODP. The intention of the paper is to raise questions for the developers of ODP standards. These questions highlight important areas of future research for both the CSCW and ODP communities.

2. THE EMERGENCE OF CSCW

2.1. What is CSCW?

The term Computer Supported Cooperative Work was initially coined by Irene Grief and Paul Cashman in 1984. Since that time, the field has grown to become one of the most prominent research areas of the 1990s. The exact meaning of the term CSCW has undergone considerable debate [1, 2]. Most authors however agree on the following principles:-

i) work is a *cooperative* activity, generally involving *groups* of people interacting to achieve common goals, and

ii) the designers of supporting computer systems must address this cooperative nature of work.

It could be argued that the above statements are obvious. However, it is clear when reviewing the current state of the art in computer technology that the cooperative nature of work has been largely ignored. For example, existing word processors provide excellent environments to help an author to produce highly professional documents and yet fail to consider the fact that most document production is collaborative.

The term *groupware* has emerged to signify software systems which address the cooperative nature of work. In the words of Lynch, Snyder and Vogel [3]:

> "Groupware is distinguished from normal software by the basic assumption it makes: groupware makes the user aware that he is part of a group, while most other software seeks to hide and protect users from each other. . . . Groupware . . . is software that accentuates the multiple user environment, coordinating and orchestrating things so that users can 'see' each other, yet do not conflict with each other."

2.2. A multi-disciplinary approach

Perhaps the most striking feature of CSCW is the inter-disciplinary nature of the work. Inter-disciplinary research in computing is not new. For example, human computer interaction has developed as an inter-disciplinary subject involving psychologists and computer scientists.

In CSCW, contributions to the field have been made by such diverse disciplines as computer science, psychology, sociology, economics and organisational studies. Perhaps the most notable contribution has come from sociological traditions. In particular, the field of ethnographic analysis has emerged as a particularly prominent technique to aid the development of groupware systems [4].

Ethnographic analysis (or ethnography) is grounded in studies of the *sociality* of work, i.e. work is fundamentally a social activity. The purpose of ethnography in not so much to show *that* work is socially organised but to show *how* it is socially organised. The roots of ethnography are in anthropology and ethnographers observe workers in their environment over an extended period. The aim is to gain an understanding of the actual as opposed to the formal work practices.

Ethnographers recognise that work is *situated* [5], i.e. work is carried out in a social context. Furthermore, within the shared context there are many subtle interactions involving gesture, chance remarks and informal interchanges that might well be as important as the more formal tasks being carried out. There is also a great deal of implicit knowledge involved in work practices. This is closely linked to the notion of a *working division of labour* [4], in that the process of allocating tasks amongst individuals can be very flexible, often based on factors such as the current context and level of activity and not necessarily on prescribed roles or procedures.

It has been argued that the key advantage of ethnographic analysis is that it captures detail which more traditional approaches to requirements analysis often miss. It is important to stress though that ethnography is not an answer to the problems of requirements capture. Rather, ethnography provides only one useful viewpoint of a system. Other viewpoints are equally valid and indeed are often complementary.

2.3. An illustrative example

As an initial example of CSCW in practice, we consider a research project at Lancaster

University which is looking at computer support in the field of air traffic control. This research focuses on the development of a prototype display generator to provide an electronic replacement for the current paper-based flight progress strip. It is now recognised that such automation is required because of the increasing pressures on UK air space.

There have been a number of attempts to automate the UK air traffic control system. However, these attempts have not been popular with air traffic controllers. Hopkins argues that the main reason for this has been the lack of awareness of the cooperative nature of work in air traffic control [6]:

> "One striking aspect of automation applied to air traffic control systems is that most of the forms of automation for the controller to use, as distinct from those which sense or process or compile data automatically, are for one controller at a human-machine interface. They are aids to an individual controller's decisions, problem solving or predictions, yet they are being introduced into contexts where many of these functions have been performed by teams".

In contrast, the Lancaster study started of with the premise that air traffic control is a highly cooperative activity and hence carried out a detailed ethnographic analysis of work practices and used this to guide the prototype development.

The ethnographic analysis highlighted the trustable and reliable nature of the existing system. Interestingly, this reliability does not stem from the intrinsic reliability of the individual components in the system (staff, equipment, procedures). Rather, the reliability stems from the system as a whole and depends largely on rich and dynamic teamwork by controllers. The study also highlighted the importance of the flight progress strip in this process. Each flight strip is a strip of card approximately 8 inches by 1 inch which contains information about expected and current flights being controlled, together with controllers' instructions to the controlled aircraft. The flight information for a new strip is derived from a database of flight plans. The initial information is then amended as instructions are given to and confirmed by the pilot.

Flight strips are organised on a *flight progress board* where strips are aligned and organised in a rack according to the reporting points over which a flight will pass. Experienced air traffic controllers can derive considerable information from this progress board, e.g. the anticipated future loading on the system or emerging problems requiring attention.

Flight strips are central to the process of air traffic control acting as a central point of coordination for air traffic controllers. Much of the time, controllers appear to be working in isolation. However, the flight strip provides a publicly available workspace which enables controllers to ‘at a glance' monitor the overall state of the system and the work of others. For example, controllers can rapidly provide assistance for colleagues if they are becoming overloaded. The flight strip also provides a public history of the state of the sector being controlled and with it *accountability* in the collective process.

In designing the prototype system, the ethnographic analysis did *not* provide detailed system requirements; essentially, it provided background information and a deeper understanding of the application domain. However, there are a number of examples where the ethnographic study guided the system design in crucial ways. For example, the study highlighted the importance of manual manipulation and manual re-ordering of the flight strips. Flight strips are ordered by their time of arrival to a beacon. It would therefore seem natural to add new strips, positioned automatically in the right place on the user's display. However, manual positioning draws the attention of controllers to the new arrival and helps to identify potential problems at an early stage. This example highlights the fact that some conventional assumptions made by system designers may be invalid when cooperative systems are being developed, e.g. that computer systems should always automate tedious tasks. Apparently straightforward tasks may have a number of significant side-effects in rich cooperative settings.

It is clear from this work that ethnography can play an important role in *informing* the process of automated systems design. However, there are a number of significant problems which must be overcome for such methods to work effectively, particularly in bridging the gaps between the two disciplines involved. The approaches of the two subjects are fundamentally different: sociology is *analytical* whereas computer science is more concerned with *synthesis*. Sociology pays a great deal of attention to detail whereas computer science strives for abstraction. Similarly, ethnographers are trained to avoid making judgements whereas engineers *must* make judgements. However, the Lancaster project has demonstrated that the two disciplines can work together effectively with some flexibility on either side. It is clear though that further study is required to understand the complex relationship between ethnographic analysis and systems analysis and design. A fuller account of the ethnographic analysis and subsequent prototype development can be found in the literature [7].

3. AN OVERVIEW OF CSCW RESEARCH

In this section, we consider the current state of the art of CSCW research in terms of the research on emerging cooperative applications (i.e. groupware) and the infrastructures to support them. This research is complemented within CSCW by a consideration of the nature of systems design, the nature of organisations and the use of systems within actual work settings. Interested readers are referred to [8,9] for more complete considerations of CSCW research.

3.1. The space-time matrix

An initial consideration of groupware systems proposed by Johansen [10] focused on the two principle characteristics of the developed application:

i) The form of interaction (*synchronous versus asynchronous)*

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

ii) The geographical nature of users (remote versus co-located)

Cooperative systems can be considered as being either *remote* or *co-located.* Note that this division between remote and co-located is as much a logical as a physical one and is concerned with the accessibility of users to each other rather than their physical proximity.

Johansen presents his characterisation of groupware as a space-time matrix for groupware (see figure 1)

	Same Time	**Different Time**
Same Place	Face-to-face interaction	Asynchronous interaction
Different Places	Synchronous distributed interaction	Asynchronous distributed interaction

Figure 1. The groupware space-time matrix.

This matrix provides a useful *initial* characterisation of groupware applications. However, some difficulties have been identified with the divisions implied by the matrix. In practice, work often switches rapidly between asynchronous and synchronous interactions. CSCW researchers now highlight the need to support these transitions in as seamless a manner as possible [11]. Recent research in CSCW has therefore focused on specific domains of application, e.g. co-authoring, and the development of support techniques spanning all parts of the space-time matrix. A selection of the most important application areas are presented below.

3.2. Prominent areas of activity

3.2.1. Workflow systems

Perhaps the most prominent class of cooperative system are those based on the notion of workflow or activity. These systems have developed from previous considerations of office automation and adopt a process oriented perspective on group work. Cooperative work is viewed as items of work flowing between a number of activities. Message exchange is the predominant means of representing this flow of work.

The current generation of CSCW applications provide models and mechanisms aimed at supporting either a particular cooperative activity or class of activities. Each of these applications postulate their own particular view of activities and how they should be structured. Some exploit formal models based on either speech act theory (e.g. Co-ordinator and action-workflow [12]) or office procedures (Domino [13]) while others adopt a considerably less formal approach(Object Lens[14]).

3.2.2. Multimedia and desktop conferencing systems

Desktop conferencing systems stem from the merging of workstation technology and real-time computer conferencing. Such systems enable groups of users to simultaneously interact with one or more applications; voice and video facilities are also often provided. Two main approaches have been identified in the CSCW community, i.e. collaboration-transparent and collaboration-aware conferencing.

Collaboration-transparent systems enable existing applications to be viewed in a group setting (examples include Rapport [15], SharedX [16] and MMConf [17]). As the application is unaware of the presence of more than one user, it is necessary to multicast display output and multidrop user input so that the application deals with a single stream of output and input events. To avoid confusion, users must take turns in interacting with the application; this is achieved by adopting an appropriate floor control policy.

In contrast, collaboration aware solutions provide facilities to explicitly manage the sharing of information, allowing sharing to be presented in a variety of different ways to different users. The management of sharing is often embedded in the application itself. Such applications are often referred to as *multi-user applications* and represent the emerging generation of CSCW applications (these include Cognoter[18], Grove[19] and Meade[7]). However, there are a number of outstanding problems with this approach. The lack of supporting infrastructure requires most collaboration aware applications to be constructed from scratch. This has held back the development of such systems. More significantly, applications tend to encapsulate the of decisions as to how information is presented and modified. This lack of visibility inhibits tailoring of the sharing policy in conferences.

3.2.3. Multi-user hypertext

Multi-user hypertext systems are a significant focus of research in CSCW because of their ability to support flexible structuring mechanisms. Within such systems, the hypertext document (or network) is constructed by a number of users adding nodes to the network in an independent manner. Facilities must then be provided to deal explicitly with the conflicts inherent in this process of interaction. Systems following this line of development include Intermedia [20] and Notecards [21]. More recently, systems such as Sepia [22] have extended the provision of support for cooperative hypertext by developing facilities to support the representation of cooperative work plans as part of the network.

Co-authoring systems are one class of cooperative systems which apply the principles of hypertext technology in a cooperative setting. The Quilt system [23] developed at Bell Communications is representative of the general principles used by most co-authoring systems. 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 is that the users read a publicly available document annotating the document to reflect their comments. At any time a Quilt comment network will consist of a current base document, some revision suggestions, and a set of comments. A similar approach to co-authoring systems is provided by the GROVE system [19] at MCC.

3.3. Emerging areas of activity

The previous section provided a small sampler of traditional areas of activity in CSCW systems development. To complement this essentially historical perspective, this section highlights some important emerging areas of work in cooperative systems development. These areas of research provide an insight to the future demands CSCW is likely to place on the supporting infrastructure provided by the distributed systems community.

3.3.1. CSCW toolkits

Developing cooperative systems is a notoriously difficult endeavour. Very little is known about how people actually cooperate and few studies have been done on identifying a set of effective guidelines for the development of cooperative systems. Consequently, designers of CSCW systems are often forced to rely on their own intuitions. These intuitions are often wrong and have prohibited the success of CSCW systems [24]. As a result, a growing focus of research has been in the development of toolkits which allow the rapid construction of applications to support a prototypical approach to systems development. Each of these has in turn constructed their own particular set of supporting services. These prototyping toolkits include:

Rendezvous [25]: an architecture and prototyping language designed to support the development of real-time interfaces for cooperative applications.

Mead [7]: a prototyping environment which supports the rapid construction of multi-user interfaces by direct manipulation.

DistEdit [26]: a development toolkit to support the construction of shared editing systems.

OVAL [27]: a Lisp based object oriented toolkit which allows the rapid construction and reconfiguration of a range of cooperative applications based upon sharing structured information.

3.3.2. The use of space

A consideration of the spatial characteristics of work is emerging as a central consideration in the development of cooperative systems. It is now recognised that spatial metaphors can be useful in supporting cooperation. For example, the concept of rooms is used extensively in user interfaces as a means of partitioning and organising work [28]. Similarly, several projects employ a virtual meeting room metaphor in computer conferencing systems, providing facilities such as personal spaces (offices), shared spaces (meeting rooms) and doors to move between such spaces [29].

More recently, a range of multimedia systems have also been developed with the intent of forming distributed shared *media spaces* across a user community [30, 31]. These systems use multimedia communication facilities to simulate the everyday environment within which people work. Perhaps the best known example is the experiment at Xerox PARC linking two coffee rooms with a shared video wall [32]. A variety other systems have been reported including Cruiser, developed at Bellcore, and the more asynchronous system Portholes [33], developed at EuroPARC. Recent work has focused on extending the ability to explore the remote spaces through techniques such as using multiple cameras with moving fields of view [30].

The essence of media spaces is to embed multimedia communication technology within the workplace to provide an *augmented reality* where the everyday features of the workplace are extended by facilities provided by computer systems. In addition to this work, a number of CSCW researchers have been examining the use of the *virtual reality* techniques to create shared artificial spaces to support cooperative work (example systems include Rubber Rocks [34] which allows shared interaction with simple objects in space, the work of Takemura which focuses on collaborative design in virtual reality [35] and DIVE [36] which features a spatial model for cooperation in large unbounded space [37]).

3.3.3. Mobility

Recent advances in portable computers and mobile communications are stimulating interest in extending support for cooperative work into the field. In modern organisations, significant numbers of workers are mobile, either travelling between sites or carrying out work at remote locations. These technological advances have the potential to revolutionise work practices for such employees. For example, the utilities industries in the UK are looking very closely at the potential impact of mobile support for field engineers [38].

At present, there is very little work in this area. Most research has a technological focus. For example, there is considerable research into communications technologies to support mixed media (voice and data) into the field. Researchers are also looking at issues such as disconnected file systems [39] and addressing mechanisms for mobile computers [40]. There has been much less work on the social impact of mobility. It is likely however that mobile computing will grow to be one of the major areas both in CSCW and in ODP.

4. IMPORTANT ISSUES FOR ODP

4.1. Enterprise/ Information Viewpoints

It is certainly true that there has been much less progress in populating the Enterprise and Information Viewpoints than, say the Computational Viewpoint. Indeed, many researchers are looking to CSCW to provide some answers in this area. They see ethnography as an important tool in understanding complex enterprises and would like to build on this in providing languages for Enterprise and Information modelling. At present, though, there is a significant gap to bridge before this becomes possible (if indeed it is possible at all). An examination of the output of an ethnographic study reveals large volumes of case notes, interviews and observations but little apparent methodology.

The problem stems from the inter-disciplinary gap refereed to in section 2.4. In general, the field of CSCW is suspicious of the use of abstraction. They view abstraction as *prescribing* a particular model on the world; in CSCW, which is attempting to model complex real-world interactions, models are therefore given careful scrutiny. To the computer science profession, such discussions are often difficult; they view abstraction as their only tool in mastering complexity. The CSCW view is perhaps best exemplified by Robinson and Bannon who in a consideration of CSCW systems development state [41]:

> "When refinements to the initial model were presented, often these refinements were also based on abstractions rather than on any clear empirical evidence for the relevance of these features in actual work situations".

Supporters of this position often quote experience with early CSCW systems such as Co-ordinator. As mentioned in section 3.1, Co-ordinator is a workflow system based on speech act theory. Effectively, speech act theory provided a model of the work practices being modelled. Experience with Co-ordinator has often been very negative [2]. Many of the problems stem from the overly prescriptive nature of this underlying model [42]:

> "Co-ordinator makes explicit and textual a dimension of human communication which is otherwise contained in the overall context of interaction. It further makes the unsupported assumption that participants in the system will willingly share the designers' view that one should be extremely explicit about the nature of one's utterances".

Indeed, such problems led to some researchers to refer to Co-ordinator as the "world's first fascist computer system". Some researchers in CSCW however now view this extreme position as being unhelpful. They see an important challenge in bridging the gap between ethnography and systems design. This work is not however explicitly examining the relationship between ethnography and ODP; this remains an interesting avenue for further research.

4.2. Computational/ Engineering Viewpoints

4.2.1. Supporting group work

Transparency vs awareness

The concept of transparency (hiding from a particular user the potential behaviour of some parts of the system) is viewed as a cornerstone of ODP standardisation. As mentioned above, the CSCW community is generally suspicious of such use of abstraction and hence transparency has come under some scrutiny. It has been argued that, in CSCW, *awareness* is often as important as transparency. This issue is particularly important when we consider concurrency transparency which focuses on the existence of concurrent activity (and by implication, other users). In cooperative applications, there is a clear requirement to be aware of other concurrent activity, especially in more synchronous styles of interworking.

Existing approaches to concurrency transparency generally rely on the concept of *atomic transactions* [43, 44]. Within such schemes concurrency transparency is achieved by *prescribing* the principle of serialisability, i.e. concurrent operations are allowed only if their combined effect is equivalent to a serial sequence of operations. This is achieved in most approaches through the use of *locks*. Such approaches give the impression of shared access being carried out in isolation of other users and hence do not meet the awareness requirements of CSCW systems. For example, consider the case of a system designed to support the co-authoring of documents. If a group member is updating a section of text, then it might make sense for an interested colleague to "read over their shoulder".

The problem with existing approaches to transactions is illustrated in figure 2a which shows a number of users accessing shared information. The approach in transaction mechanisms is to control this shared access by creating walls between the different users and the existence of other users is masked out completely. This contrasts with the requirements of CSCW applications where information flow between users is as important as interaction through the shared information space. This information flow between users enables a social protocol to be established to regulate access to shared information. This approach is illustrated in figure 2b.

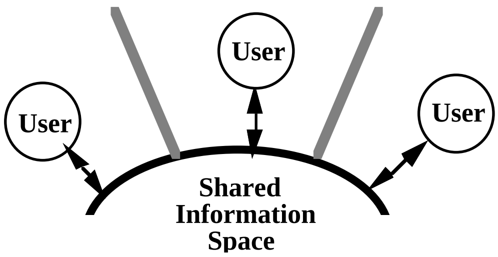

Figure 2a. The transaction approach.

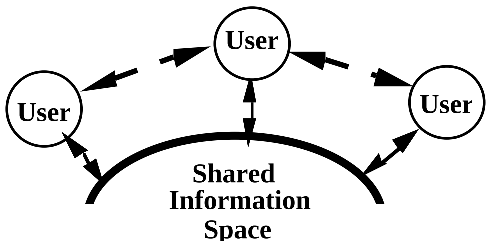

Figure 2b. An alternative approach.

A further problem with existing transaction mechanisms is the lack of support for *real-time responsiveness*. Ellis highlights two real-time requirements for CSCW applications: response time, which must be short to support a highly interactive system, and notification time, the time it takes for one user's actions to be propagated to the other users in a group session [19].

Several researchers have investigated the provision of transactions for group working. This research is still at an early stage. However, some interesting results are starting to emerge. For example, a number of researchers have proposed alternative styles of locking to increase the flexibility of transaction mechanisms, e.g. *tickle locks* [45], *soft locks* [18] and *notification locks* [46].

Although such schemes do provide some support for groupware applications, a number of problems exist. For example, it is not clear in joint authoring applications whether locks should be applied at the granularity of sections, paragraphs, sentences or even words.

A number of more radical schemes have also been proposed. For example, Skarra and

Zdonik [47] have introduced the concept of a *transaction group* which co-ordinates access to shared data for a number of co-operating members. Within a transaction group, the notion of serialisability is replaced by access rules based on the semantics of the cooperation. Access rules provide the *policy* of cooperation and these policies can be *tailored* for a particular application by amending the access rules. In addition, the group editor GROVE [19] adopts a new form of concurrency control based on *operation transformations*. This allows operations to proceed immediately to improve real-time response time. To maintain consistency, it might be necessary however to execute a transformed operation rather than the original operation. Other real-time applications have tackled the issue of concurrency control through the use of *reservation*. Conferencing systems often use a floor passing approach to reservation. Other systems, such as Colab, use an approach based on more informal negotiation. Reservation is only suitable however for approaches that do not want to interleave operations.

A more recent trend has been to address this problem from the opposite direction and provide explicit *awareness mechanisms* for both synchronous and asynchronous modes of working. This work often uses spatial and temporal metrics to generate awareness weightings defining the impact of actions on other users. Visualisation techniques are also being developed to provide appropriate levels of awareness. Mariani[48] describes a prototype implementation of a collaborative object store, based on an extension of an organisational knowledge base browser.

In summary, it is clear that traditional approaches to transactions are not well suited to group work. In particular, they prescribe a solution to concurrency transparency which is not suitable for the styles of interaction required by CSCW. A number of alternative mechanisms are emerging which provide more tailored support for CSCW. In the context of ODP, some research is required to determine whether such proposals can be accommodated within the current framework of transparencies.

Security

Most of the discussion in the CSCW community has focused on the *access control function* of security. Most existing approaches to access control in distributed systems are based on the classic Access Matrix. Specific mechanisms derived from this matrix include access control lists and capabilities.

A number of criticisms of this general approach have been made in the CSCW community [49, 8, 45]. It is now generally recognised in CSCW that access control policies should be based on the concept of *role*. Furthermore, it is recognised that roles are dynamic, changing frequently during the course of a collaboration. In contrast, most existing mechanisms are based on identification of an individual with the assumption that this identity does not change. In addition, most existing access models adopt a static view of the management of access control information. The assumption is that access is sent up and only occasionally altered by a single administrator. However access models within CSCW system should also support *dynamic* changes to access control information. It is also likely that such changes will be made as a result of *negotiation* between parties involved. Suggestions have also been made that access control should be at a *fine level of granularity* [49], for example constraining the access to individual lines of a shared document.

The potential added complexity of access control in a CSCW environment raises some interesting issues in terms of *specification* of access rights. It is important in CSCW environments that access rights are both visible and easy to understand in order to accommodate frequent change.

There has been little research addressing the provision of access control mechanisms for groupware applications. Shen and Dewan however describe a novel scheme featuring fine grain control and multiple dynamic user roles [49].

Management

Current ODP proposals support a range of management functions including node management, capsule management, cluster management, communications management and configuration management. It is important that these functions take into account the particular requirements of CSCW. The most important issues identified to date are that of the initial placement of objects (node management) and their subsequent re-location (cluster management). The mapping of objects to clusters/ capsules/ nodes is crucial for the performance of groupware applications. This is particularly true for synchronous applications requiring real-time response. In such applications, objects are likely to be shared by a group of users at geographically dispersed sites with each site requiring similar real-time response. This adds considerable complexity to the placement and migration strategies of objects.

There has been very little practical experimentation in this area. However, it is clear management functions must be aware of the pattern of use of objects emanating from groups. In more general terms, group aware *policies* are required. This also assumes that appropriate *mechanisms* are in place to support and inform such policies.

4.2.2. Additional challenges

Multimedia support

As seen in section 3, a significant number of groupware applications require sophisticated multimedia support, e.g. virtual and augmented reality applications. It is now recognised that multimedia computing imposes a range of new challenges to ODP both in terms of computational and engineering viewpoints:

i) Support for continuous media

The most fundamental characteristic of multimedia systems is that they incorporate continuous media [50] such as voice, video and animated graphics. Continuous media (e.g. video and audio) have an implied temporal dimension, i.e. they are presented at a particular rate for a particular length of time. If the required rate of presentation is not met, the integrity of these media is destroyed. In ODP, there is the requirement to *represent* continuous media in a computational model (e.g. through the introduction of *streams*) and, secondly, there is the requirement to *support* continuous media data types in the engineering viewpoint, (through appropriate protocol selection).

ii) Quality of service

The timeliness of media transmissions is maintained through quality of service (QoS) parameters, e.g. throughput, end-to-end delay (or latency) and delay variance (jitter). Again, this requirement has implications for both the computational and engineering viewpoints. In the Computational Viewpoint, it is necessary to support the expression of desired levels of QoS. In terms of engineering, quality of service support must be provided on an *end-to-end* basis from the information source to the information sink. This includes the network and communications subsystem and also the operating system and multimedia devices. Facilities are required for negotiation of QoS levels between remote peers and also for end-to-end monitoring of QoS so that the application can be informed if degradations occur. Dynamic re-negotiation should also be supported.

iii) Real-time synchronisation

Multimedia applications require an extensive range of real-time synchronisation mechanisms to control the precise timings of interactions between separate multimedia activities. In analysing the requirements of multimedia applications, two styles of real-time synchronisation can be identified [51]: firstly, *event driven synchronisation* where it is necessary to initiate an action (such as displaying a caption) at a particular point in time and, secondly, continuous synchronisation, where data presentation devices must be tied together so that they consume data in fixed ratios (e.g. in lip synchronisation). The provision of real-time support has an impact on both the Computational and Engineering Viewpoints.

iv) Groups

In CSCW, it is clearly important for a multimedia distributed system architecture to support the concept of mixed media group interaction. In the computational viewpoint, there is a need to support *group communications* for continuous media transmissions. This style of interaction is used, for example, if a video source is to be displayed in a number of distinct video windows simultaneously. There is also a requirement to support group invocation, for example if a group of cameras are to be started simultaneously in a conference. The requirement for *group invocation* is already being discussed in the ODP forum although current proposals will need to address real-time requirements imposed by groupware applications. In terms of engineering, multicast transport protocols are necessary to enable group communication of continuous media. In addition, group RPC protocols are required which provide bounded real-time performance.

The ODP community has recently responded to these challenges by proposing a number of extensions to the draft standards. For example, extensions have been made in terms of stream interfaces and stream bindings. The draft standards also include text on quality of service annotations of interfaces. It is clear however that further work is required to fully meet the requirements of multiparty multimedia communications. For example, further research is needed to identify approaches for the expression of quality of service properties and compatibility checking between these properties. In addition, engineering support is required for *end-to-end* quality of service management. It is also clear that ODP standardisation has not yet fully addressed the problem of supporting real-time synchronisation. Similarly, important issues remained unresolved in supporting group communication (particularly at an engineering level). There is a body of relevant research however in the multimedia community to build on. For example, joint research between Lancaster University and CNET has recently proposed a Computational and Engineering model to meet the requirements of continuous media communications [51].

The impact of mobility

With the relative immaturity of mobile computing, it is difficult to assess its likely impact on ODP. Several potential problem areas can however be highlighted:

i) Techniques for transparency

ODP must address the issue of providing transparent access to services from mobile hosts. With the limited bandwidth of radio communications, this means that new techniques will be required, for example, to cache significant portions of the data on the mobile computer. Care must also be taken to maintain consistency if data is shared across several mobiles, e.g. in a conference situation.

ii) Impact of disconnection

Mobile communications are characterised by their peculiar error characteristics, for example users are likely to be disconnected for significant periods of time. It is important that quality of service requests can specify accepted levels of disconnection and that quality of service management can monitor and react to such circumstances. It is also important to trace the impact of disconnection on other dependent services.

iii) Levels of disconnection

Over a period of time, connection may vary from being disconnected to being partially connected (through a radio network) to being fully connected (through a high speed network). This is likely to have a significant impact on techniques for configuration management and binding support. It is also likely that services will take advantage of higher levels of connection to perform bulk updates, e.g. of cached data.

iv) Communications architectures

It is clear that new protocols are required to cope with the characteristics of mobile communications. Particular attention is required by the need to support multicast communications across radio systems. New techniques are also required for mixed media traffic. Existing mobile communications architectures are tailored towards the support of voice traffic. They are not well suited to the bursty style of data traffic generated by many distributed applications.

The impact of mobility for ODP is currently being investigated in the MOST Project (Mobile Open Systems Technologies for the Utilities Industries) involving Lancaster University, EA Technology and APM.

5. CONCLUDING REMARKS

This paper has examined the particular challenges of CSCW for Open Distributed Processing. The major results of this investigation are summarised below:

i) the need to consider the sociality of work in Enterprise and Information Modelling

ii) the need to avoid overly prescriptive languages across all viewpoints

iii) the need for more cooperation aware policies for concurrency control, security and management

iv) the need to provide sophisticated support for multiparty multimedia communications

v) the need to address the emerging requirements for mobile computing

Many of these areas require further study (especially the first and last issues). It is hoped this work can lead to a more substantial dialogue between the two communities to address these outstanding problems.

ACKNOWLEDGEMENTS

We would like to thank our colleagues at Lancaster University for their helpful comments and interesting discussions on the subject matter of this paper. Special thanks are due to John Hughes, John Mariani, Graham Dean, Dick Bentley, Nigel Davies and Lynne Blair.

REFERENCES

1. Bannon, L. and Schmidt, K. (1991) 'CSCW: Four Characters in Search of Context' in J.M. Bowers and S.D. Benford (eds): Studies in Computer Supported Cooperative Work. Theory, Practice and Design, North-Holland, Amsterdam.
2. Robinson, M. (1991) 'Computer Supported Cooperative Work: Cases and Concepts' Proceedings of Groupware'91, Software Engineering Research Centre, Postbus 424, 3500 AK Utrecht, Nederland, pp 59-74.
3. Lynch, K.J., Snyder, J.M., Vogel, D.R. and McHenry, W.K. (1990) 'The Arizona Analyst Information System: Supporting Collaborative Research on International Technological Trends', in Gibbs, S. and Verrijn-Stuart, A.A. (eds): Multi-User Interfaces and Applications, North-Holland, Amsterdam, pp159-174.
4. Hughes, J.A., Randall, D. and Shapiro, D. (1992) 'Faltering from Ethnography to Design', Proceeding of CSCW'92, Toronto, November 1992.
5. Suchman, L. (1987) 'Plans and Situated Action: The Problem of Human Machine Communication', Cambridge University Press.
6. Hopkin, V.D. (1991) 'The Impact of Automation on Air Traffic Control Systems', in J.A. Wise, V.D. Hopkins and M.L. Smith (eds.), Automation and Systems Issues in Air Traffic Control, Berlin, Springer Verlag.
7. Bentley, R., Hughes, J., Randall, D., Rodden T., Sawyer, P., Sommerville, I. and Shapiro, D. (1992) 'Ethnographically Informed Systems Design for Air Traffic Control', Proceeding of CSCW'92, Toronto, November 1992.
8. Ellis, C.A., Gibbs, S.J. and Rein, G.L. (1991) 'Groupware: Some Issues and Experiences', Communications of the ACM, 34(1): pp 38-58.
9. Baeker, R. (1993) 'Readings in Groupware and Computer Supported Cooperative Work', Morgan Kaufmann, San Mateo CA, ISBN 1-55860-241-0.
10. Johansen, R. (1988) 'Groupware: Computer Support for Business Teams', New York, The Free Press, Macmillan Inc.
11. Ishii, H. (1990) 'TeamWorkStation: Towards a Seamless Shared Space', Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90).,Los Angeles, CA, October 7-10 1990, ACM press , ISBN 0-89791-402-3, pp 13-26.
12. Medina-Mora, R., Winograd, T., Flores, R. and Flores, F. (1992) 'The Action Workflow Approach to Workflow Management Technology', Proceeding of CSCW'92, Toronto, November 1992.

13. Kreifelts, T., Hinrichs, E., Klien K., Seuffert, P. and Woetzel, G. (1991) 'Experiences with the DOMINO Office Procedure System', Proceedings of ECSCW 91, Amsterdam, September 1991, Kluwer Academic Press, pp 117-131.
14. Lia, K.-Y. and Malone, T.W. (1988) 'Object Lens: A 'Spreadsheet' for Cooperative Work', Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '88), ACM Press, pp 115-124.
15. Ahuja, S.R., Ensor, J.R. and Horn, D.N. (1988) 'The Rapport Mulimedia Conferencing System', in Allen R.B. (ed), Proceedings of the Conference on Office Information Systems (COIS’88), March 1988, Palo Alto, California.
16. Garfinkel, D., Gust, P., Lemon, M. and Lowder, S. (1989) 'The SharedX Multi-user Interface User's Guide, Version 2.0.', Report STL-TM-89-07, Hewlett-Packard Laboratories, 3500 Deer Creek Road, Palo ALto, CA 94304.
17. Crowley, T., Baker, E., Forsdick, H., Milazzo, P. and Tomlinson, R. (1990) 'MMConf: An Infrastructure for Building Shared Applications', Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90),Los Angeles, CA, October 7-10 1990, ACM press , ISBN 0-89791-402-3.
18. Stefik, M., Foster, G., Bobrow, D., Kahn, K., Lanning, S. and Suchman, L. (1987) 'Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings' Communications of the ACM, 30(1): pp 32-47.
19. Ellis, C.A. and Gibbs, S.J. (1989) 'Concurrency Control in Groupware Systems', In Proceedings of the ACM SIGMOD International Conference on the Management of Data, Portland, Oregon, ACM Press, pp 399-407.
20. Garret, L.N., Smith, K. and Meyrowitz, N. (1986) 'Intermedia: Issues, Strategies and Tactics in the Design of a Hypermedia Document System', Proceedings of CSCW’86, Austin, Texas, December 1986.pp 163-175.
21. Trigg, R., Suchman, L. and Halasz, F. (1986) 'Supporting Collaboration in Notecards', Proceedings of CSCW’86, Austin, Texas, December 1986.
22. Haake, J.M. and Wilson, B. (1992) 'Supporting Collaborative Writing of Hyperdocuments in SEPIA', Proceeding of CSCW'92, Toronto, November 1992.
23. Fish, R.S., Kraut, R.E., Leland, M.D. and Cohen, M. (1988) 'Quilt: A Colloborative Tool for Cooperative Writing', in Allen R.B. (ed), Proceedings of the Conference on Office Information Systems (COIS’88), March 1988, Palo Alto, California.
24. Grudin, J. (1989) 'Why Groupware Applications Fail: Problems in Design and Evaluation', Office: Technology and People, 4(3): pp 245-264.
25. Patterson, J.F., Hill, R.D., Rohall, S.L. and Meeks, W.S. (1990) 'Rendezvous: An Architecture for Synchronous Multi-user Applications', Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90), Los Angeles, CA, October 7-10 1990, ACM press , ISBN 0-89791-402-3.
26. Knister, M.J. and Prakash, A. (1990) 'DistEdit: A Distributed Toolkit for Supporting Multiple Group Editors', Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90), Los Angeles, CA, October 7-10 1990, ACM press , ISBN 0-89791-402-3.
27. Malone, T.W., Lai, K.-Y. and Fry, C. (1992) 'Experiments with Oval: A Radically Tailorable Tool for Cooperative Work', Proceeding of CSCW'92, Toronto, November 1992.
28. Henderson, A. and Card, S. (1985) 'Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention', ACM Transactions on Graphics, 5(3), July 1985.
29. Cook, S. and Birch, G. (1991) 'Modelling Groupware in the Electronic Office', International Journal of Man Machine Studies, 34(3): pp 369-394.
30. Gaver, W., Moran, T., MacLean, A., Lovstrand, L., Dourish, P., Carter, K. and Buxton W., (1992) 'Realising a Video Environment: EuroPARC's RAVE System', In Proc. CHI '92 Human Factors in Computing Systems, Monterey, Ca., USA, May 1992.
31. Root, W.R. (1988) 'Design of a Multimedia Vehicle for Social Browsing', Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW'88), Portland, Oregon,ACM Press, pp 25-38.

32. Olson, M.H. and Bly, S. A. (1991) 'The Portland Experience: A Report on a Distributed Research Group', International Journal of Man Machine Studies, 34(2): pp 211-228.
33. Dourish, P. and Bly, S. (1992) 'Portholes: Supporting Awareness in a Distributed Work Group', In Proc. CHI'92, May 3-7, 1992, Monterey, Cal, ACM Press. pp541-549.
34. 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. and Turk, G. (1992) 'Interactive Simulation in a Multi-Person Virtual World', In Proc. CHI'92, May 3-7, 1992, Monterey, Cal, ACM Press.pp 329-335.
35. Takemura, H. and Kishino, F. (1992) 'Cooperative Work Environment using Virtual Workspace', Proceeding of CSCW'92, Toronto, November 1992.
36. Fahlén, L.E., Brown, C.G., Stahl, O. and Carlsson, C. (1993) 'A Space Based Model for User Interaction in Shared Synthetic Environments', Proceedings of InterCHI'93, Amsterdam, April 1993, ACM Press.
37. Benford, S. and Fahlén, L., (1993) 'A Spatial Model of Interaction in Large Virtual Environments', Proceedings of ECSCW'93, Milan, September 1993, Kluwer Academic Press.
38. Davies N., Blair G.S., Friday A., Cross A.D. and Raven P.F. (1993) 'Mobile Open Systems Technologies for the Utilities Industries', Presented at the IEE Colloquim on CSCW Issues for Mobile and Remote Workers, London, U.K., March 1993.
39. Kistler, J.J. and Satyanarayannan, M. (1991) 'Disconnected Operation in the Coda File System' Proc. Thirteeenth ACM Symposium on Operating Systems Principles, Asilomar Conference Center, Pacific Grove, CA, 1991, ACM Press, Vol. 25, pp 213-225.
40. Bhagwat, P. and Perkins, C. (1993) 'A Mobile Networking System based on Internet Protocol (IP)', Proceedings of USENIX Conference on Mobile and Location-Independant Computing Symposium, Boston, Mass., August 1993, pp 69-82.
41. Robinson, M. and Bannon, L. (1991) 'Questioning Representations', Proceedings of ECSCW'91, Amsterdam, September 1991, Kluwer Academic Press.
42. Carasik, R.P. and Grantham, C.E. (1988) 'A Case Study of CSCW in a Dispersed Organization', Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems, Washington D.C., May 15-19,ACM Press. pp 61-66.
43. Bernstein, P.A. and Goodman, N. (1981) 'Concurrency Control in Distributed Database Systems', ACM Computer Surveys, 13(2): pp 185-221.
44. Spector, A. (1989) 'Distributed Transaction Processing Facilities', In Mullender, S. (ed), Distributed Systems, Addison-Wesley, New York.
45. Greif, I. and Sarin, S. (1987) 'Data Sharing in Group Work', ACM Transactions on Office Information Systems, 5(2): pp 187-211.
46. Hornick, M.F. and Zdonik S.B. (1987) 'A Shared, Segmented Memory System for an Object-oriented Database', ACM Transactions on Office Information Systems, 5(1): pp 70-95.
48. Mariani J. and Prinz W. (1993) 'From Multi-User to Shared Object Systems - Awareness about Co-Workers in Cooperation Support Object Databases', Proceedings of GI Jahrestagung '93, Springer.
47. Skarra, A. and Zdonik, S. (1989) 'Concurrency Control and Object-Oriented Databases', in Kim, W. and Lochovsky, F.H. (eds), Object-Oriented Concepts, Databases, and Applications, ACM Press Frontier Series. Addison-Wesley, Pages: 395-421.
49. Shen, H. and Dewan, P. (1992) 'Access Control for Collaborative Environments' Proceeding of CSCW'92, Toronto, November 1992.
50. Anderson, D.P., Tzou, S.Y., Wahbe, R., Govindan R. and Andrews, M. (1990) 'Support for Continuous Media in the DASH System', Proc. of the 10th International Conference on Distributed Computing Systems, Paris, May 1990.
51. Coulson G., Blair G.S., Stefani, J.B., Horn, F. and Hazard, L. (1993) 'Supporting the Real-time Requirements of Continuous Media in Open Distributed Processing', Internal Report MPG-93-10, Lancaster University, January 1993.