Supporting CSCW Applications with an Efficient Shared Information Space

Erik Wilde
Swiss Federal Institute of Technology (ETH Zürich)
Computer Engineering and Networks Laboratory (TIK)
CH – 8092 Zürich
wilde@tik.ethz.ch

Abstract

Today most CSCW systems are built on top of standard operating systems. Only a few frameworks for generic support of CSCW applications exist. These platforms mostly concentrate on the management of workflows and on the layer on top of them, the CSCW applications. Little work is done in exploring the impacts of new networks onto support for CSCW. The project described in this paper focuses on providing CSCW applications with an efficient shared information space. Efficiency in this context means the utilization of network technology which offers much better services than today's networks.

1 Introduction

The development of CSCW from a field only known by a few experts to a discipline of basic and interdisciplinary research as described by Schmidt and Bannon [12] shows that, in the future, CSCW applications will be of increasing importance. While today most collaborative applications are developed as stand-alone systems which only use standard operating system support (such as transport services provided by a stack of communications protocols or window system toolkits), in the future an increasing number of application frameworks will be developed which can be used as a base for CSCW applications. Schmidt and Bannon also point out that support for CSCW applications may be divided into two general categories.

- Supporting the management of workflows

  Besides the 'normal' workflow, which is normally predefined by the steps required to fulfill a given task, collaborative work needs another level of interaction with the computer to successfully work on a given task. Robinson [10] calls this the requirement for a "double-level language". The second level of interaction is necessary to coordinate and synchronize the users of collaborative applications. The two levels may be seen as one providing the means for more "formal" communications, the other one being appropriate for "cultural" communications.

- Supporting the management of a common information space

  While the management of workflows allows users to communicate in order to coordinate their work, another type of support is necessary to provide these users with access to the same information. This common information space (or shared information space, as it is called in this paper) is needed to share the same information among a group of collaborative users. It is important to notice the difference between a shared view of information objects and a shared information space, the latter providing the user with information about the context of objects with relation to collaborative tasks.

While these two categories are of great importance to research into support for CSCW applications, this paper will only deal with the second, mainly for two reasons. Firstly, most of the research in CSCW today is done in the direction of supporting the management of workflows. Only few projects also deal with the other issue, which is therefore less developed than the first one. This hypothesis will be shown in the following section, which deals with the motivation of the approach described in this paper. Secondly, this paper is only about one part of an ongoing project which will eventually also deal with the support of the management of workflows, but which is yet in its first phase.

Furthermore, our main interest is the application of new developments in the field of networks for cooperative work. While more traditional, asynchronous systems do not have many requirements on the network (except reliability and a reasonable speed), modern synchronous CSCW applications are much more demanding. There is a need for multimedia capabilities, perhaps for remote playback of audio and video. These requirements make the utilization of new networks for CSCW support very promising.

2 Motivation

While the development of cooperative applications becomes more and more popular, most of the projects implementing these programs only use standard operating system support. Only a few toolkits have been developed until now, some of them are listed in the following section. These systems do not use special network support, they mainly concentrate on other issues, such as preservation of context or sharing of distributed objects.

Rodden, Mariani and Blair [11] claim that a paradox has developed between the concept of the virtual machine common in today's operating systems and the need for cooperation. This is only partially true because the 'protective walls' between the different users of a machine are there to regulate access to resources which often have to be shared between a number of users. However, for the development of cooperative applications, means must be provided to allow interaction between users in a controlled way.

It is therefore necessary for CSCW research that models are created and toolkits are developed which can serve as a base for a number of CSCW applications. This not only decreases the efforts for creating applications, it also enables the interaction of these applications on a higher semantic level because of the common interface. Finally, the usage of modern computer networks providing true multimedia capabilities will increase the number of potential uses of CSCW applications.

2.1 Current work

Current work on support systems or environments for CSCW mostly concentrates on supporting the management of workflows. Examples for this type of support are the JVTOS project [4] or BERMMC [1], which are both projects whose goal is to enable a number of users simultaneously working with one application. While this sort of application sharing is ideal for using single-user applications in a multi-user context, it is too limited to be appropriate for all types of cooperative applications.

Benford, Smith, Shepherd and Howidy [3] describe a system which uses OSI services to create a *Group Communication System (GCS)*. The transport system is the X.400 standardized mail system. Although this approach fits well into today's communication infrastructure, it is limited because of the use of existing mail systems. Data transport is asynchronous, hence there is no support for real-time data.

ORION-2 [8] is a commercially available system which consists of a shared and a number of private databases. Although the architecture is well-suited to support cooperative work, there is no specific assumption about the underlying network. Consequently, isochronous transmissions of real-time data is not possible, nor is the exploitation of new features of modern networks, such as multicast and accounting mechanisms.

MultimETH [9] is a system which uses real-time audio communications, but only outside the shared workspace which is available for sharing documents. Although MultimETH claims to use a somewhat generic means of communication between the system's components using a *Multimedia Service Element (MMSE)* according to other common application service elements defined within the ISO/OSI model of open communication, it would be very hard to use this service element in another application because of its tight connection to the rest of the software. Furthermore, the assumptions about the underlying network are very conservative.

2.2 Extending the information space

While cooperative work can also be done without sharing information on the computer system level (i.e. by only transmitting pointer locations or cursor movements, as it is implemented with application sharing programs), the sharing of information is essential to cooperative work that should not suffer from severe limitations. As described by Greif and Sarin [6], the sharing of information is one of the key issues in supporting group work.

Today, sharing information also means sharing of multimedia objects in a distributed environment. There is a close connection between the development of high-speed networks and CSCW because of the requirements of cooperative work. While many of today's conferencing systems provide video transmission, there is a lack of systems supporting video as a data type which may be stored and shared. Real-time data types such as video and audio are difficult to handle in today's networks because of bandwidth problems and bottlenecks. However, future's networks will provide users with guaranteed qualities of service for data transmissions and will therefore make possible a more flexible way of handling real-time data.

3 Structure of the paper

After analyzing the requirements of CSCW applications in section 4, section 5 will give an overview of some of the problems related to the implementation of a shared information space. Because of our special interest in the issue of modern networks, section 6 describes some of the features that will be needed for the implementation of an efficient shared information space. Section 7 describes the future work of this project. This paper then will be finished with some concluding remarks in section 8.

4 Applications' requirements

This section will discuss the implications of the needs of CSCW applications on the design of a shared information space. It does not take into account the utilization of a specific transport medium.

When designing a shared information space, the focus is on providing CSCW applications with as much relevant information as possible. All information that is concerned with groups and their relations to each other and to pieces of information should be accessible to applications. This means that every application can choose which amount of CSCW related information is necessary to fulfill its task. We plan to implement a few sample applications on top of the shared information space and we will therefore see which information is most important and which is not necessary at all. This will give us a better understanding of the real requirements of CSCW applications.

4.1 Flexible groups

The notion of the group is the central issue in CSCW. While most operating systems provide support for some sort of grouping mechanism, this often is too restricted to fulfill the needs of CSCW applications.

What is needed is a group concept which is recursive, i.e. groups may consist of groups and/or users. Because grouping is essential in CSCW applications, the creation, modification and deletion of groups should be easy and efficient. The concept of the owner of information (as present in the UNIX filesystem) has to be extended with groups, too. Consequently, any piece of information may be owned by a user or a group. A user may be in as many groups as necessary, there should be no limit.

This concept of groups allows hierarchic structures (by using recursively defined groups) and overlapping groups (by using multiple group memberships). The concept of groups is closely related with confidentiality and authenticity, which are discussed in section 4.3.

4.2 Flexible access

While audio and video are becoming more and more common data types, there still is the question of how to deal with the amount of data generated by microphones or video cameras. Storing a video sequence of several minutes may require some gigabytes of disk space. This space may not be available on every system participating in a cooperative activity. However, it should be possible to show the video there, too. Remote playback of real-time data types is the solution to this problem. Because of the increasing usage of real-time data in computer applications and the problems related to their volume, access to any piece of information should be possible either locally or remote.

It is not quite clear whether the actual playback should be part of the model of the shared information space or be left outside. An alternative for integrating synchronous transmissions into the model would be the use of references. According to the model defined in the DOR standard [5], it is possible to transfer references to data objects within a given framework, while the transfer of the data itself lies outside the scope of the model. This would be a good way to ensure stability (references can be understood by every system) and flexibility (synchronous transfer of real-time data could be accomplished in a number of ways).

4.3 Security

Security within CSCW is necessary in different forms. Schmidt and Bannon [12] describe what forms of security are necessary in CSCW applications. Together with the concept of groups described in section 4.1, the following properties should be available for both individual users and groups.

- Confidentiality

  Very often data is not intended to be read by everybody. Confidentiality of information makes it possible to limit the number of users having access to specific pieces of information.

- Integrity

  Integrity is the property of a given data object which assures that it is not modified without sufficient access rights. If integrity is not guaranteed, data created by one user may be modified by another one and lead to false assumptions about the originator of the information.

- Authority

  Authorization is an important issue in any multi-user system. It describes the possibility to restrict certain operations (which create, modify, or delete information) to a given set of users and/or groups. Only by successfully identifying as an authorized person one is allowed to perform any restricted operation.

- Authenticity

  Authenticity is a property of operations which guarantees the obligatory nature of any manipulation. One example for this property is the well-known electronic mail. Most mail systems of today can not guarantee that a message is originated by the user which is listed as originator. Consequently, electronic mail can not be used for concluding contracts.

Any generic CSCW support which fails to provide these security issues will fail because of the lack of being able to support trustable applications. Today's cryptographic models are sufficient to implement all these properties, which are often tightly linked.

However, because some of these security features sometimes are necessary and sometimes not, they should be available but not mandatory. Security as a set of optional features is important for the free flow of information which is crucial to many CSCW applications. Anonymity may be appropriate in early stages of cooperative work while later phases of a project often require a thorough documentation of who did what and a guarantee that no-one else made any undocumented modifications.

4.4 Optimal distribution

While, from a very abstract point of view, the actual distribution of information may not be of any interest to applications, as long as it is available on request, this point of view may cause problems in implementations. In the real world, information is neither transmitted without any delay nor without being charged for the transmission. Applications using a shared information space therefore have an interest in having control over these two aspects of the distribution of objects. However, this control must provide for an easy way to specify the applications' requirements. It is therefore considered how applications may request their requirements with respect to the distribution of information within the shared information space.

The two subjects discussed in the following sections are deeply interrelated, because a greater efficiency in form of faster accesses usually increases costs and low costs usually degrade the efficiency because of greater transmission delays. The calculation of the optimal strategy in terms of efficiency and costs lies within the responsibility of a shared information space, which should assure the best quality of service possible. From the applications' point of view, both requirements have to be fulfilled as good as possible, no matter how contradictive they may be. Section 5.1 will discuss the interrelation of the two issues.

4.4.1 Efficiency

Efficiency of transmissions may be seen from two points of view. Firstly, for asynchronous transmissions, the highest possible bandwidth will be the best because of a shorter transmission delay. Therefore, bandwidth is the only interesting property of the transmission (except the reliability of a transmission). Secondly, for synchronous transmissions, which may be necessary for remote playbacks as described in section 4.2, some other properties such as a guaranteed transmission rate, a guaranteed transmission delay, and a low delay jitter may be of importance.

While a CSCW application may not be interested in specifying parameters for single transmissions, it may be important to have a short delay when working with data or to be able to remotely playback audio or video.

These requirements must be supported by a shared information space.

4.4.2 Costs

On the other side, each transmission is charged with specific costs. This means that the costs of transmissions also are important. In case of asynchronous transmissions, it may be tolerable to wait a little time if this saves a considerable amount of money. Most of today's networks in the research community are used without directly paying for it, consequently this aspect of data transmission is often neglected.

Furthermore, storage space also needs to be integrated into the cost model. While storage space may be very cheap for small information objects, it will become an issue with audio and video sequences of considerable size.

5 Implementation aspects

This section will deal with some aspects of the implementation of a system for the support of an efficient shared information space. There are many ways to implement a system which provides the services described in the previous section. We will describe some techniques which will be useful in the realization of an efficient shared information space, although some of the problems still are under research.

5.1 Efficiency vs. costs

The contrasting nature of efficiency, which should be maximized, and costs, which should be minimized, directly leads to a problem which may be solved using optimization theory. Both parameters, efficiency and costs, should be modelled as functions which describe how much of the respective parameter is desired and how much is absolutely necessary to be able to provide the service requested. A trade-off usually will sacrifice some of the efficiency against cheaper communication costs.

An optimal strategy will try to minimize transmissions, perhaps using caching techniques or predictive methods. Bacon [2] describes a number of strategies for this problem. However, special attention is needed for the remote playback, which is not included in most models of distributed information. Another point neglected in most models are the costs of storage. If storage is included in the optimization, the outcome may be different because of the high costs (or even impossibility) of storing large quantities of data on particular systems.

5.2 Hyperdocument techniques

In the present stage of the project, there is no model for the structure of data within the shared information space. Simply using a database which may be queried may be too less to satisfy CSCW applications. Information generated and modified by cooperative groups may be seen as a web of individual pieces, linked to each other by context information which can be crucial for correct interpretation. Consequently, the system to be designed should support this structure of information, which is best done with hyperdocument techniques.

By using attributed links, this structure may be applied for different purposes, such as describing the relation to other information objects, describing the relation according to group membership, and describing relations which may have application-defined semantics. We believe that is it not possible to describe all semantic relationships between information objects, therefore there is a need for being able to use defineable relations. The disadvantage with application-specific relations is the lack of exchangeability with other applications.

5.3 Security

Security issues are of great importance if CSCW applications should be used in all phases of a project. The security model employed in most operating systems, based on authorization (logging in with a password) and permissions for individual objects, may be extended to match the needs of the shared information space. For example, permissions can no longer be seen as a fixed set, since group membership is recursive and not limited to one group. Furthermore, ownership should not be limited to individuals. Information may also directly belong to groups without one specific individual being its owner.

However, to assure confidentiality, integrity and authenticity, other methods have to be used. There are some cryptographic techniques which make these properties possible, but they need special protocols to exchange keys and encrypted data. Because of these requirements, security issues have to be investigated very close before defining any protocol for the exchange of information.

6 Network requirements

This section highlights some of the features which should be provided be the supporting network. Because of the requirements of the applications, the network needs to provide more services than today's networks. The following issues are not absolutely necessary to make the shared information space possible, but their lack will influence performance and quality of service offered by the shared information space. Some of the features, e.g. real-time data transmissions, may even be impossible.

The term network is used for simplicity. In fact, the shared information space will not use any network directly, but a transport service layered on top of a network. Consequently, the requirements listed here are to be fulfilled by a transport service, which in turn uses specific (perhaps several) networks. However, some features, such as multicast which is discussed in the following section, need to be provided by the underlying network in order to work efficiently.

6.1 Multicast

CSCW applications are among the most popular uses for multicast networks. While traditional networks and communication services only support point-to-point communications (and sometimes broadcast mechanisms), newly designed networks are able to deliver data sent by one user to a number of other users.

There are differences in how groups that may be addressed are defined, depending on the specific multicast model. What is needed in case of the shared information space is a flexible and easy way to define groups, i.e. to add and delete members of groups and to dynamically create new groups. Optimal support would be possible with a recursive group concept which could be used to directly map the recursive group concept of the shared information space onto it, but this requirement may be to demanding to be satisfied by the networks of the near future.

6.2 QoS parameter

The properties of the shared information space discussed in sections 4.4 and 5.1 could only be implemented in a reasonable way if the underlying network provides the informations necessary to calculate the efficiency of data transfers. This can only be done if a set of quality of service parameters is defined and can be used for connection setup.

QoS parameters to be provided by the network should include the following items, additional parameters may increase the accuracy of efficiency calculations of the shared information space. The exact form of QoS specifications (integer values, ranges, functions) depends on the QoS model used within the network. However, it should always be possible to specify the following values.

- Throughput

  The most important property of a connection is the throughput (or bandwidth). There is a great difference between maximal throughput and guaranteed throughput. A large maximal throughput is good for bulk data transfers, which can also tolerate times of much smaller throughput. For synchronous data transfer, there has to be a guaranteed throughput to match the needs of the data transferred.

- Costs

  It is important to know the costs of data transfers. Although a large maximal throughput may be best in almost all cases, this may be too expensive. Consequently, it must be possible to specify a maximal amount of money one is willing to pay for the transfer.

- Transit delay

  When using synchronous data transfers it is important to know how long the data will be in transit. Also in asynchronous cases, e.g. when data is transferred in a store-and-forward manner (which is normal for mail systems), it may be necessary to specify an upper bound for the delay tolerated. If the delay is too large, it may become necessary to use caching techniques.

- Delay jitter

  This parameter is only of interest for synchronous transmissions. It specifies the maximal variation of the synchronicity. A large delay jitter may force a system to use large buffers (which increase the delay), therefore a maximum has to be guaranteed to make real synchronicity possible.

Much work is done in the area of QoS parameters. It is important to have a close look on the parameters available in new networks and to analyze their relevance for a shared information space. The parameters listed above may only be a subset of what is required to be able to control the transmissions of data as much as necessary. Jain and Agrawala [7] give a good classification of a greater number of parameters.

6.3 Accounting mechanisms

While QoS parameters are used to specify the requirements for connections, it must also be possible to get accounting information after the transfer of data is completed. This information (including transmission characteristics according to the QoS parameters) may then be used to modify the QoS parameters or even the transmission strategy for subsequent transmissions. For example, if it is seen that data transfers usually are very expensive, the distribution strategy may be modified to cause as few transmissions as possible.

7 Further work

Further work on the shared information space will include a number of steps. Firstly, we will define a formal model for the shared information space which will probably be based on graph theory. Modelling the different entities as nodes and the communication lines as edges gives a clear view of the topology of the shared information space. Many of the parameters of the system, such as network QoS parameters and accounting parameters of the entities (such as storage costs) can then be modeled as attributes of the graph's elements.

Secondly, we will define an information model which can be used to formally define the structure of the information stored within the shared information space. A hypergraph model will be used to accomplish this, giving us the freedom to define any structure which can be used in a hyperdocument information model.

In a third step, we can use the model for simulations. Using available graph simulation tools, we can test different distribution strategies and analyze the impact of different utilization patterns on the behaviour of the shared information space. This simulation will make it possible to choose a good strategy for building a prototype.

Fourthly, a prototype will be implemented which may be used as a test tool for building sample applications. By using the prototype for implementing CSCW applications, we will be able to see which aspects of our model are the most useful, which ones are less used and what is missing. This will lead us to the conclusion of the project, which will be a comprehensive set of properties required for the efficient implementation of a shared information space.

8 Conclusions

A model for a shared information space has been described which concentrates on using new network technology to work as efficient as possible. It has been shown that some of the features of these networks, such as multicast and QoS parameters, are very useful in supporting CSCW applications. An implementation of an efficient shared information space could use these features as well as cryptographic techniques and optimization methods for the distribution of information.

References

[1] Michael Altenhofen, Jürgen Dittrich, Rainer Hammerschmidt, Thomas Käppner, Carsten Kruschel, Ansgar Kückes, and Thomas Steinig. The BERKOM multimedia collaboration service. In *Proceedings of ACM Multimedia 93*, pages 457–463, Anaheim, California, 1993. ACM Press.

[2] Jean Bacon. *Concurrent Systems*. Addison-Wesley, Wokingham, England, 1992.

[3] Steve Benford, Hugh Smith, Alan Shepherd, and Howidy Howidy. Information sharing in CSCW: From models to architecture. In *Proceedings of the International Workshop on CSCW*, pages 232–251, Berlin, April 1991.

[4] Gabriel Dermler, Thomas Gutekunst, Bernhard Plattner, Edgar Ostrowski, Frank Ruge, and Michael Weber. Constructing a distributed multimedia joint viewing and tele-operation service for heterogeneous workstation environments. In *Proceedings of the Fourth Workshop on Future Trends of Distributed Computing Systems*, pages 8–15, Lisbon, 1993. IEEE Computer Society Press.

[5] International Organization for Standardization (ISO). Part 2: Distinguished object reference (DOR) and transfer model. Proposed new text for ISO/IEC DIS 10031-2, April 1990. Information technology — Text communication — Distributed-office-applications Model (DOAM).

[6] Irene Greif and Sunil Sarin. Data sharing in group work. *ACM Transactions on Information Systems*, 5(2):187–211, 1987.

[7] Bijendra N. Jain and Ashok K. Agrawala. *Open Systems Interconnection: Its Architecture and Protocols*. Elsevier, Amsterdam, 1990.

[8] Won Kim, Nat Ballou, Jorge F. Garza, and Darrell Woelk. A distributed object-oriented database system supporting shared and private databases. *ACM Transactions on Information Systems*, 9(1):31–51, 1991.

[9] Hannes Lubich. *MultimETH: Ein Beitrag zur Konzeption eines Echtzeit-*

Multimedia-Konferenzsystems. PhD thesis, Swiss Federal Institute of Technology, Zürich, 1990.

[10] Mike Robinson. Double-level languages and co-operative working. *AI & Society*, 5(1):34–60, 1991.

[11] T. Rodden, J. A. Mariani, and G. Blair. Supporting cooperative applications. *Computer Supported Cooperative Work*, 1(1–2):41–67, 1992.

[12] Kjeld Schmidt and Liam Bannon. Taking CSCW seriously. *Computer Supported Cooperative Work*, 1(1–2):7–40, 1992.