

Document-Centric Groupware for Distributed Governmental Agencies

Daniel A. Tietze^{*}, Ajit Bapat^{*}, Rolf Reinema^{**}

GMD – German National Research Center for Information Technology

^{*} Integrated Publication and Information Systems Institute (IPSI)

^{**} Institute for Telecooperation Technology (TKT)

{daniel.tietze | ajit.bapat | rolf.reinema}@gmd.de

Abstract

The distribution of the German government between Bonn and Berlin calls for the technical support for the collaborative document-based tasks performed by inter- and intra-ministerial work groups. The currently available cooperation and communication tools do not provide an integrated and homogeneous cooperation environment suitable for this application domain. This paper presents the Cooperative Document Repository, an integrated document-centric collaboration environment. The integration of cooperative document management and document processing with multi-point audio/video conferencing provides a flexible collaboration environment within which a wide range of collaborative activities can be performed.

Keywords: groupware; CSCW; document-based collaboration; federal government; distributed administrations; POLIWORK; video conferencing

Introduction

The German unification and the resulting move of governmental agencies from Bonn to the new capital, Berlin, will result in a distribution of administrative units between Bonn and Berlin. This distribution will be both along vertical as well as horizontal lines, meaning that some ministries will be moved entirely to Berlin, some will remain in Bonn, while others will be distributed between Bonn and Berlin. As a result, work groups and task forces (both within a ministry as well as between ministries), which were previously co-located will be distributed as well and will require technical support in order to continue their work. To address the challenges of this distribution and to provide technical solutions and organizational suggestions for the resulting distributed work processes, the German government instated the POLIKOM research program in 1994, within which four projects are addressing the different issues arising from the close cooperation between distributed governmental agencies.

At GMD, the authors are involved in one of these projects: POLIWORK¹ – Telecooperation and cooperative document management at the personal workplace – which began in 1994 and will continue until the end of 1998. The focus of this project is to support the cooperative tasks of the governmental agencies, specifically those collaborative activities centering on document management and document processing, both in synchronous as well as in asynchronous settings, through the development and installation of appropriate support technology. The user community of the project consists of two German federal ministries, the Ministry of the Interior (BMI) and the Ministry of Economics (BMW).

This paper describes the central software component developed for the user community: the Cooperative Document Repository (CDR), an integrated collaboration support system which inter- and intra-ministerial work groups can use in order to structure and store the documents involved in their cooperative work processes as well as to initiate the required communication and collaboration tasks. The remainder of the paper is organized as follows. The next section presents brief scenarios of collaborative activities found in the user community. From these a set of requirements is derived to be addressed in the subsequent system design. The architecture and functionality of the CDR are then presented, followed by a description of the approach of document-centric collaboration support. The paper concludes with a comparison to related work in the field of collaboration support technology and a brief outlook on future work.

System Requirements

The requirements for the technical solution which are presented in this section were gathered in the user community through an analysis of the cooperative work processes found as well as through interviews with the users. The resulting work process scenarios were analyzed in order to determine how to adequately support the arising tasks by introducing a collaboration support system combining state-of-the-art communication hardware with advanced groupware technology. Also, the existing infrastructure found within the ministries was examined in order to determine in which way it could be integrated into a collaboration support environment.

Collaborative Activities

In the scope of the project, we concentrated on two main scenarios for cooperative activities in the ministerial environment: joint text creation by a work group as well as presentation and discussion including the higher levels of the ministerial hierarchy.

A typical task in the application field is the preparation of a legislative draft. Usually, such a text is created jointly by a small group of people. Different parts of the document are created by different authors and then integrated into one common

¹ The work presented in this paper is partially funded by the German Federal Ministry of Education, Science, Research and Technology (grant number 01 IT 403 A-D).

document. The various intermediate versions of the draft are passed around among the members of the work group. Paper documents are distributed using fax or regular mail while documents in electronic form are sent via e-mail. To coordinate the integration of the various parts bilateral communication (e.g. telephone call) as well as multilateral communication (e.g. meeting) takes place. Meetings include a certain amount of preparation (setting a date, distributing the necessary documents, etc.) as well as wrapping up (writing minutes etc.).

Another scenario is closely related to the one described above. At the end of the joint text creation the resulting draft needs to be approved by the higher levels in the ministerial hierarchy. For this, the final draft is passed to the higher level and then, in a meeting with the superior the draft is presented and discussed, changes to be made are annotated. Afterwards, the document is revised accordingly.

Requirements

Based on the work processes found in the ministerial context, a number of requirements can be identified for a system that is to support the distributed, cooperative tasks in the application field.

(R1) All work processes are document-based. Either texts are created and edited as part of the process or the process involves the presentation, discussion and annotation of documents. Therefore, *document-based support* is indispensable. This requirement combines several important aspects: Documents need to be *integrated* directly into the work processes, including the support of existing document bases (R1.1). In order to work on the documents, *cooperative document processing* is required. One important aspect of this is the support of legacy applications (R1.2), since the ministries have a certain set of document processing tools which they regularly use (e.g. one of the established office suites). Documents have to be passed to other people involved in the task. Thus, *document exchange* is to be supported (R1.3). Furthermore, *storage of the documents* is necessary (R1.4). And finally, *paper-based documents* need to be integrated (R1.5).

(R2) The second important aspect is the provision of *communication* channels between the people involved in the work processes. *Synchronous* communication (R2.1) and the *high quality* of the communication channels (R2.2) are the major requirements. Since the collaborative processes span multiple sites, the communication support should not only be bilateral but also support *multi-party distributed conferencing* (R2.3).

(R3) One key aspect with respect to the acceptance of the system by the users is the *usability* of the supporting system. To achieve this, several aspects have to be considered: A *homogeneous* user interface (R3.1), *integrated access* to various functionalities of the system (R3.2), as well as a *familiar look and feel* of the system's user interface (R3.3).

(R4) Due to the fact that the system is to be used in the Federal Ministries the existing technical infrastructure found there needs to be taken into account. The ministries are connected by an *ISDN* network which is physically separate from the

public telephone and ISDN networks and therefore regarded as being secure. Thus, communication should run over these lines (R4.1). In addition, the government is setting up a *governmental IP backbone* that connects all ministries within and among the two cities of Bonn and Berlin. Again, this network is to be used (R4.2). Additionally, the system that is to be developed should run on the *governmental intranet* which is currently being installed (R4.3).

(R5) In addition to the previous requirements, some general requirements result from the fact that the system is developed in a pilot project. *Adaptability* to other user fields, e.g. other ministries (R5.1), as well as *scalability* to larger numbers of users (R5.2) are required.

System Design

In order to ensure rapid availability of the key functionalities required in this project a major design goal was to take already existing components from the market. By doing so, an initial set of requirements could be fulfilled by off-the-shelf components. These application modules then needed to be integrated into the system in a way which makes the combined functionality of the tools easily available to the users. A first version of a collaboration support environment has already been installed at an early stage so that the users could become familiar with the new possibilities offered by communication and cooperation technology.

Support for joint viewing and editing of documents by two or more participants is provided by the use of a standardized application sharing package which enables several users at the same time to jointly view and interact with any application and thus any kind of document. In this way, the standard application suites and legacy applications which were already used within the ministries as well as the documents produced by these are integrated into the collaborative processes in a familiar and easy to use manner. In addition to this the use of a color flatbed scanner made it possible for the users to import paper based documents and enabled them to work on them. Both the sharing of any application between several sites as well as the possibility to import any paper-based document fulfilled the requirement for joint viewing and editing of existing document bases.

As already mentioned above, the users have asked for high quality synchronous audio/visual communication while working together on documents. This resulted in the provision of off-the-shelf components for audio/video and data conferencing. The two user communities within the project, the German Federal Ministry of Economics (BMWi) and the Federal Ministry of the Interior (BMI) have had different needs for communication and collaboration support [1]. While the BMWi has expressed the need for desktop-based collaboration between single users in the form of desktop workstations equipped with conferencing and collaboration technology, the BMI has asked for meeting rooms to let groups of users from within the ministry or between different ministerial agencies collaborate with each other. So they were provided with a number of electronic meeting rooms (so called team rooms) equipped with large projection units coupled to the same kind of

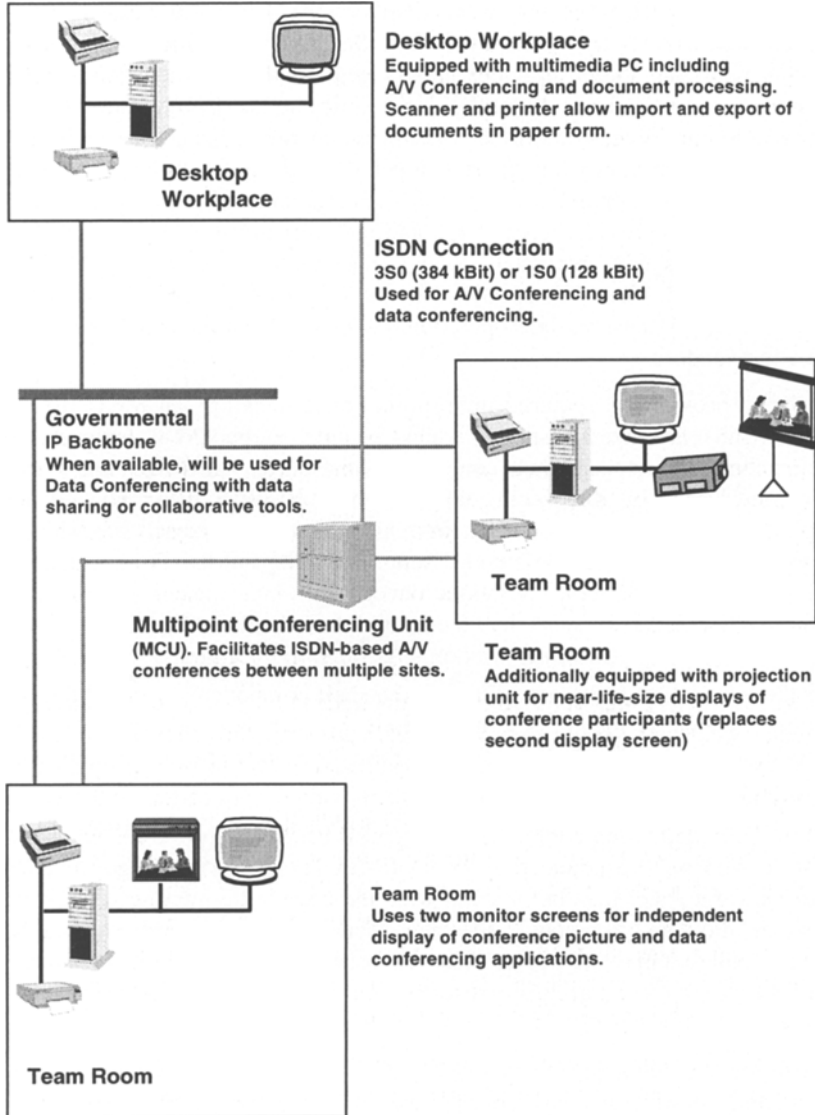


Figure 1 : Variations of technical solution

workstations as already used for the desktop users. Different configurations achieved in the initial solution are shown in figure 1.

As a basis for the audio/visual communication there were two different kind of networks: a governmental owned ISDN network (BBN) as well as a governmental owned IP-network (IP-Backbone). Because of the fact that currently every ministry

has access to the BBN but not to the IP-Backbone, ISDN has been chosen as the basis for synchronous communication. A further and by far more important reason was the fact that ISDN provides guaranteed bandwidth and delay which are necessary to provide the audio and video quality the users have asked for. For the desktop solution, a bandwidth of 128 kbps was deemed appropriate, while the large projection unit in the team rooms, together with the requirement of hosting discussions between a number of users at each site required higher bandwidths (384 kbps), in order to provide better video and audio quality for all users in the room. Hardware Codecs with integrated ISDN connectivity are used in order to achieve high performance. Their compliance with the international ITU-T standards H.320 [2] and T.120 [3] ensures interoperability with ISDN-based conferencing systems from other vendors.

In order to provide the required multi-point communication, an MCU (Multi-point Control Unit) has been installed which is able to handle a large number of simultaneous conferences, each between several sites. The MCU manages multi-point conferences by distributing the A/V data between the connected sites. It integrates multiple A/V data streams into a single stream for each site which could, for instance, display multiple sites on one screen, e.g. in the form of a quad-split display showing four remote sites at the same time.

Assessment of Initial Solution

After the installation and setup of the off-the-shelf components some of the previous mentioned needs of the users could already be met, but there were still several remaining requirements which necessitated the development of the system presented in this paper.

With the provided video conferencing system it became possible for all users (even for those who had no connection to the IP-Backbone) to exchange documents in electronic form faster than before. But what the users were missing was a document repository into which they could file their documents for retrieval within as well as between collaborative sessions. The repository should allow structuring the document base by work groups and restrict the access to those work groups or task forces.

Many of the necessary steps to establish collaboration between two or more users are purely administrative or consist of application access and application control, tasks which the integrated collaborative application could easily perform for the users. Additionally, many of the necessary steps are performed by different applications, each providing its own distinct user interface, forcing the users to be familiar with a large number of applications just in order to perform the task at hand. Instead, the envisioned system should be provided with enough information beforehand, so that arranging a collaboration session between a number of users would be ideally reduced to merely selecting the document(s) to be cooperatively processed and the user(s) with whom to collaborate. All administrative tasks, the scheduling and reservations as well as the timely invocation of the required tools were to be

controlled by the system, thus giving the users the freedom to concentrate on the main task at hand: the joint processing and creation of documents.

A major requirement of the users was the provision of a single integrating user interface to control all the different functional modules of the application suite. The off-the-shelf components were neither integrated nor did they provide a homogeneous user interface with a familiar and consistent look and feel to the users.

A general problem with today's MCUs is that they do not offer any kind of (standardized) API (Application Programmers Interface), rather they require an operator who carries out conference reservation and control instead of the users. This resulted in the requirement to develop a conference reservation and control system which allows users to create and control audio/video and data conferences themselves. Another problem with today's MCUs is that ad-hoc communication is almost impossible, because of the way in which an MCU conference is set up: When a user decides that he needs a multi-point conference he has to ask an MCU operator to set it up at a certain day and time. This can be very time-consuming (it may take several minutes to hours depending on the availability of the operator). Also, the way an MCU controlled conference is typically set up results in the problem that a multi-point audio/video conference can not be resized dynamically in terms of time and number of participants, i.e. during a running conference, no further users can participate unless that their later participation was already anticipated when making the necessary reservation.

An analysis of the remaining unmet requirements led to the specification and development of a Cooperative Document Repository (CDR), into which the initial components were to be integrated and which provided a consistent and easy to use interface to the available functionalities.

Architecture of the Cooperative Document Repository

The CDR is a distributed collaborative system, providing the users with a cooperatively accessible document storage as well as immediate information about other users' actions. The architecture is centered around a database server which provides persistence of documents and data objects as well as propagation of real-time update notifications. This central database is provided using a commercially available RDBMS accessed via the ODBC [4] database access layer. The use of a standard RDBMS and ODBC as an access interface was chosen in order to meet the adaptability requirement, as it does not limit the system to a specific DBMS and thus ensures applicability in a wide range of user communities.

The central database stores all information required by the system: the document management metadata, the infrastructure and group modeling data as well as the actual document contents. For the Cooperative Document Repository a central data model was developed, based on previous work on a file and document model for the public administration [5] performed within the project. The original object model was revised to address the specific requirements introduced by the collaborative settings to be supported by the project's application goals, especially the stronger

need for group and access right modeling. The CDR object model contains all relevant information about not only the contents of the document repository, but also about the system environment, the users accessing the repository and the conferencing equipment used at the different sites.

The client applications of the distributed system were developed in Java, using the JDBC [6] interface as a means to access the database server. Java was chosen as an implementation language in order to make use of the existing and future infrastructure within the ministries. The use of the Java programming language and the JDBC database access mechanisms ensures operability of the solution over the governmental IP backbone and also enables, at a later stage, the integration of CDR modules as general services into the currently evolving governmental Intranet. The use of the standard database access mechanism JDBC provides a large degree of flexibility in the selection of the database system used within the system as well as the actual driver library used for database access. The database access module is configurable in order to allow easy exchange of database drivers, which also aids the evaluation of the performance characteristics of the different access libraries. As an example for this flexibility, two different JDBC packages were used interchangeably and evaluated in the course of the development of the CDR.

The truly cooperative use of the CDR in the context of a synchronous audio/video conference necessitates the synchronous coupling between the users' client applications. This is performed through instantiation of the CDR object model in the clients in the form of a distributed shared memory. The clients' data sets are automatically kept consistent through the use of the underlying database's transaction system and a combination of object invalidation and active notifications. Client side operations are gathered into transactions, communicated to the database server, whose responsibility it is to ensure transactional consistency. Objects modified within a transaction are identified and an active notification component, based on trigger mechanisms available in the database server, automatically distributes the changes to all connected clients by invalidating the locally cached object replicas and notifying the client applications of these changes. The automatic notification leads to a redisplay of those changed data objects which are currently displayed on the user's screen, thereby providing interactive synchronous collaboration.

The client applications contain interfaces to external components such as A/V conferencing, application sharing and integration of a scanner for documents in paper form. Many of the off-the-shelf components used in the project, such as the application sharing system or the videoconferencing system, merely provide C++ programming interfaces which had to be made available to Java applications in a consistent manner. In order to ease flexibility and exchange of components at a later stage, generic Java interface modules were specified for the individual functional modules which abstracted from the actual programming API and presented the module's functionality in an abstract and reusable manner. These interface modules were then mapped onto the interfaces available in the application modules currently used. This abstraction layer fulfills the adaptability requirement, since it effectively

hides the set of third-party cooperation and communication tools actually used and facilitates easy exchange of system components.

In order to fulfill the requirement for supporting existing document processing applications and enable the cooperative work on documents created and maintained with non-cooperative applications such as the familiar office suites, the CDR provides an integration with external applications (e.g. a text processing system) as well as with third-party application sharing packages. This is achieved through an abstract application integration layer which can be tailored to support a wide range of applications for different document types as well as a number of application sharing packages, accessed through their API functionality. The integration with the application sharing package allows automatic launching of the appropriate application along with the application sharing module within a conference. Due to its knowledge of the current interaction situation and the set of connected users, the CDR can launch the relevant tools and applications in a mode which is required for the current communication setting.

To support the requirement of integrating paper-based documents into the cooperative work processes, scanners and printers are integrated into the CDR. Documents can be scanned and directly imported into the CDR by way of a TWAIN compatible scanning device, thus making them directly available to the other collaboration partners.

For multipoint conferences a reservation has to be made on the MCU. The functionality for making such an MCU reservation is integrated into the CDR. The control and scheduling of the Multipoint Conferencing Unit is performed via the MCU's reservation database, since the MCU itself does not offer a directly usable API. For this task, a specialized daemon was developed which mediates between the MCU's reservation database (resident on a dedicated server machine) and the CDR database. Any relevant changes performed in the CDR database (e.g. the scheduling of a new conference) is immediately fetched and transferred to the MCU reservation database. This daemon interfaces to the CDR server just like the other clients, by way of the distributed shared memory which is automatically kept in a consistent state when changes occur, thereby informing the daemon that changed data needs to be propagated to the MCU reservation database.

The architecture diagram in figure 2 shows the "building blocks" of the CDR application. The shaded boxes denote the modules developed in Java, boxes with emphasized borders denote individual applications within the CDR.

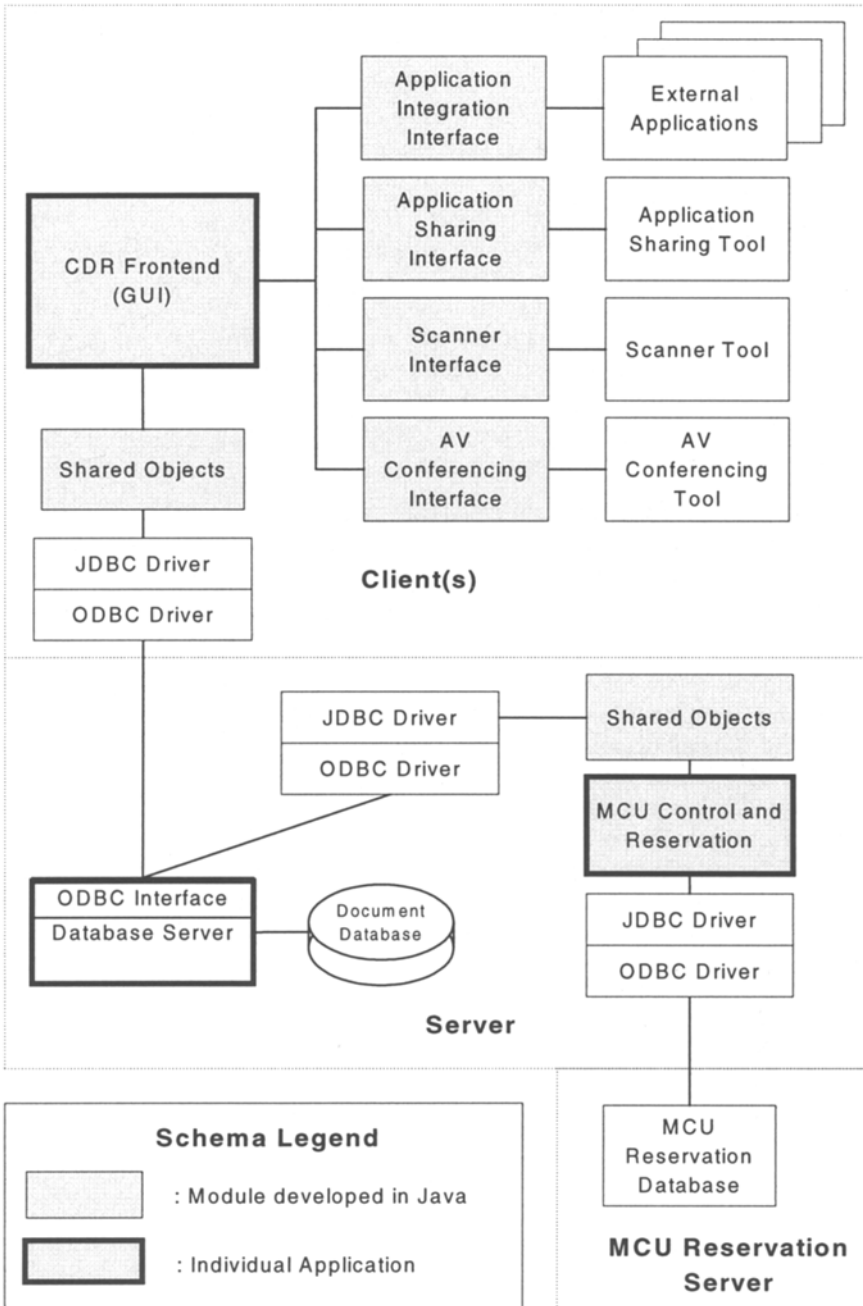


Figure 2: Architecture Diagram of CDR

Usage and Functionality of the Cooperative Document Repository

The repository front-end is a group-aware application: it not only provides the users with a view of the repository's contents, but also displays information about which users are currently cooperatively viewing or editing which documents. This group-awareness is provided in the form of name tags added to the repository content displays, indicating current cooperation activities on the documents (see figure 3).

In order to be immediately applicable by the users, the CDR provides a document management paradigm similar to the file management structures already familiar to the users from the Windows environment. The CDR enables the users to structure the cooperatively used document space into individual repositories, containing a hierarchy of folders, which in turn contain the documents. The access rights and group modeling mechanisms present in the CDR allow the users to control and manage their group cooperation and restrict other users' access to the documents in the CDR.

Every user of the Cooperative Document Repository can create groups and repositories to match his or her current requirements. For a task which is to be tackled cooperatively, a new group within the system can be created, including those users which require access to the common documents. Along with this, a new repository can then be set up, using group- and user-based access permissions to control which users and which user groups may view or edit the documents contained in the repository. The newly created repository (or, rather, an icon depicting it) automatically appears on the CDR desktops of all users who are allowed to access the repository and who are currently logged into the system, allowing them to begin their work immediately. The solution is kept scalable to a large number of users by only presenting those repositories and repository contents at the GUI to which the current user has access. The group members can now cooperatively begin to structure and fill the repository, by creating folders as required and importing documents relevant for the task at hand. In this way, the group members are provided with a common, cooperative document storage system which they can use to store and exchange documents related to the task at hand. The fact that all users have concurrent access to the CDR simplifies exchange of documents.

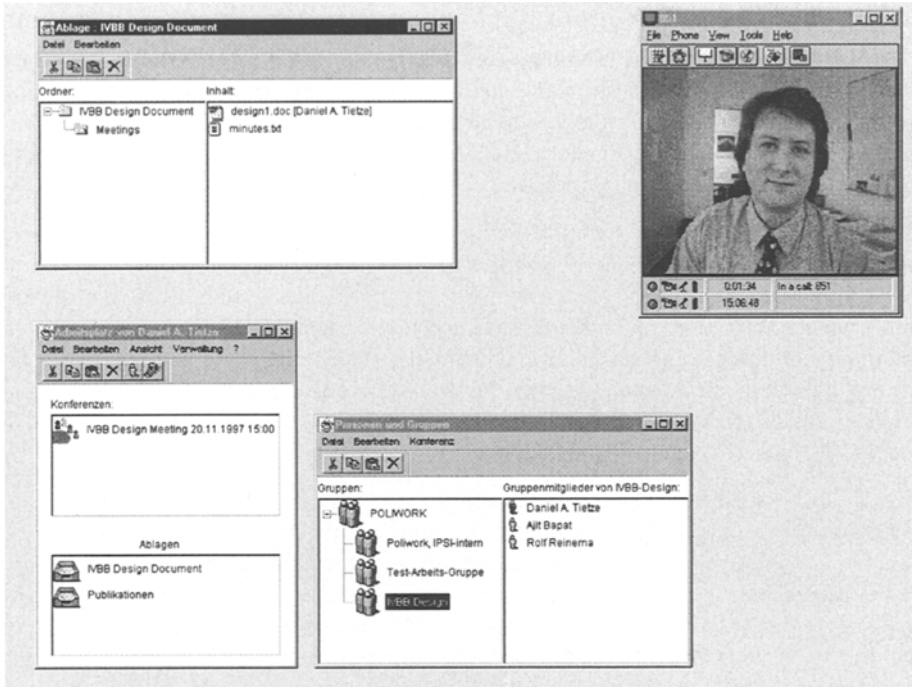


Figure 3 : Desktop view of CDR

It is important to note that the CDR was not intended to be a replacement for classical document management and storage systems. All tasks involved in the strict document management methods required in the ministerial context, such as registering all documents entering and leaving the ministry, managing consistent document and file numbers, maintenance of a filing plan, etc. are outside the scope of the CDR. All official document management functionalities will continue to be provided by dedicated document management systems (DMSs), which are approved for use in the ministerial environment and which potentially differ from ministry to ministry.

The structuring means and functionalities provided by the CDR are driven by the requirements of the task-oriented and flexible cooperation activities within the user community. The CDR does not enforce a strict organizational model of the individual ministries, departments, etc. The cooperation structures encountered in the application domain do not rely on these organizational models but instead involve work groups spanning organizational bodies and set up for short- to medium-term activities. Also, the CDR is used as the unifying basis for more informal cooperations such as unplanned ad-hoc activities. These informal collaboration tasks would more often be hindered than simplified by adherence to too strict an organizational model.

Document-Based Collaboration

To give an impression of the functionality of the CDR and to provide an illustration of how the integrated system facilitates the cooperation processes, a short scenario is presented in the following.

Suppose a user is member of a working group which is to come up with a document on the security of firewall technology for protecting the Governmental IP backbone. Assume that this user has been given the task to prepare an initial draft for this document. The cooperative scenario could, e.g., look like this:

1. In the CDR the user opens the repository that has been set up for the working group in question. With a double-click on the document the associated external application (here: a text processor) is started from within the CDR and the user can carry out his task of preparing a draft.
2. In the course of his work he might reach a point where a direct discussion with other members of the working group becomes necessary. From the list of group members the user selects his collaboration partner(s), indicates to the system that the document he is currently working on is to be subject of the collaboration and then simply invokes a „collaborate“ command from a pop-up menu within the CDR.

The CDR then automatically performs all steps which are necessary to set up this document-based ad-hoc conference:

- a) An audio/video conference is set up between all collaboration partners. To enable a multipoint conference a corresponding conference is registered on the MCU. The MCU can then call all participants and establish the multipoint A/V conference. All data necessary for this technically complex step is available in the system (the telephone numbers, the types of the involved end points, etc.). After they accept the call, the video window is finally opened on each of the users' desktops.
 - b) The application sharing package that is needed to cooperatively edit the document is started for every participant in the conference. Here again, all technical information (e.g. connection numbers) is available in the system and is used to set up the sharing session without any manual interaction by the users. The text processor that is to be used cooperatively appears on the desktops of the conference participants.
3. Now that the working group has come together, the draft preparation can continue. The contents can be discussed over the audio/video connection and changes or extensions can be entered by any of the participants.
 4. Users can easily input paper documents to the system by using the integrated scanner solution. The scanned document is directly imported into the group's document repository. Documents can of course also be printed if wanted.

5. The participants can leave the conference at any time individually. All related connections are closed automatically and finally the whole conference is shut down.

This example illustrates that working on documents in the repository as well as seamlessly switching from single user work to multilateral cooperative tasks requires only a minimum of user actions while the system can perform all complex „supporting actions“ automatically. In the following, some features of the CDR are presented in more detail.

Initiation of Cooperative Sessions

In order to cooperatively process a document from the CDR together with other users of the system, ideally all the user needs to indicate to the CDR desktop is which document he wishes to process together with which other users. Since the system is provided with all the necessary information about the users' infrastructure, such as the phone numbers of the ISDN videoconferencing system available to the user, the supported bandwidths, etc., it is able to schedule and establish a collaboration session without further user input – apart from the obvious interaction steps of confirming the communication partners' readiness to enter the collaboration session now or to postpone it to a later point in time. Since the documents which form the basis for the collaborative activity are already present in the CDR, distribution of these documents also becomes an easy task.

Conference Scheduling

A reservation system is available which generates all technical entries required by the MCU database and manages reservations of the required team rooms. A small part of the data needs to be input by the user (names of the participants, the date, time and duration of the conference etc.), while a major part of the information is stored in and retrieved from the CDR database (e.g. telephone numbers for the A/V connection, type of A/V hardware). Thus, an operatorless operation of the MCU is possible. Should no MCU resources be available for the chosen time slot, the user will be immediately informed that the conference is not possible at the chosen time. In addition to the MCU resources, any team rooms that are involved are checked for availability and are then reserved for the new conference. The newly created conference automatically appears in the conference list on the desktops of all conference participants, thus informing them about the new conference.

When the time of the conference arrives, the MCU either automatically dials out to all participants or the CDR notifies the user about a conference being started. As soon as the user has confirmed his readiness the appropriate connections are established (again without requiring the users to remember and dial any numbers etc.). The initiation of ad-hoc conferences is achieved in the same way, simply by scheduling the start of the conference with the current date and time.

Integration of External Applications

Opening a document within a conference will indicate the user's desire to present this document to the entire group of connected users or to edit it cooperatively with them. The CDR supports this activity by automatically launching not only the required application but also the application sharing component – if necessary, on all connected machines – and bringing the document's application into the application sharing session ready for collaboration. This support relieves the users of a large part of the burden involved in setting up the conference connections, the required applications and the application sharing system. Should a document be editable or presentable by a truly collaborative software such as the DOLPHIN [7] system, which does not require the additional application sharing support, this integration is also available in the CDR. Again, since the CDR has access to all the necessary information about the collaboration situation, the external collaborative application can be invoked on all sites, receiving the required parameters for connection and setup from the CDR.

Summary

Using the integrated collaboration environment provided by the CDR in conjunction with the integrated communication and collaboration tools as presented above, the users can initiate and conduct their collaboration tasks in a manner which is appropriate to the problems they actually need to tackle. Indicating the collaboration partners and the document(s) on which to collaborate is sufficient for the system to set up the required communication channels, make the required reservations and start the necessary tools at the right point in time. The users are relieved of the effort of having to control (and learn) a number of separate applications and can concentrate on the actual work processes.

Related Work

A number of research and development projects have taken on the task of supporting document-based collaboration between multiple distributed sites. This section compares representatives of these systems to the requirements listed in this paper and addressed by the CDR.

The BSCW shared workspace system [8] is a Web-based groupware application which follows a collaboration paradigm similar to the CDR: users can create and structure repositories into which they can import documents which are then made available to all users who currently have access to the repository. The documents can be retrieved by the users, processed and at a later stage put back into the BSCW system for the other users. As a purely Web-based application it differs from the CDR in a number of points which were of importance in our application context (as shown in the requirements section). Firstly, BSCW is an asynchronous system which does not provide direct feedback and information about other user's actions (synchronous group awareness; R2.1). In order to observe changes within the repository, the Web page has to be manually reloaded by the user. Also, BSCW provides no integration of synchronous multi-party A/V communication (R2.2 and

R2.3). The CoopWWW [9] project aims at extending the BSCW system through the integration of A/V conferencing between the system's users. It currently does this by integrating a software A/V communications module, CUSeeMe [10]. The resulting system does not offer the high-quality A/V conferencing supplied by the CDR with the integrated ISDN conferencing system. Since CUSeeMe is a software solution and employs the Internet for data transport, the available bandwidths and thus the achievable frame rates and picture quality are much lower than the ones achieved by integrating ISDN-based multipoint conferencing as we have done in the CDR. An evaluation of user requirements has shown, though, that audio and video quality are of utmost importance for the acceptance and wide-spread use of the resulting technical solution (R2.2).

Lotus Realtime Notes [11] is a combination of Intel ProShare and Lotus Notes. Notes, as a replicated distributed document database can be used to store documents which are accessible to a group of users. The use of Intel ProShare allows real-time communication and collaboration on these documents by use of the application sharing component. What this combination does not achieve, though, is the seamless integration of multipoint communication and multi-party distributed settings (R2.3). Also, this combination does not provide an integrated user interface with a familiar look-and-feel (Requirements R3.1, R3.2 and R3.3). The CDR goes one step further towards a complete integration by integrating scheduling, control and reservation of multiparty conferences. Furthermore, Realtime Notes does not provide synchronous group awareness to the system's users.

Neither Realtime Notes nor BSCW/CoopWWW are adaptable to different standards-based collaboration and communication tools in the way in which the CDR is (R5.1).

The TeamRooms system [12] also aims at supporting collaboration of distributed teams by providing persistent repositories for the collaboration documents. TeamRooms, though, does not integrate synchronous multimedia conferencing (R2.1, R2.2, R2.3) and uses specialized "applets" for cooperative document creation, thus has no provision for legacy applications and seamless integration into the collaboration processes (R1.1, R1.2).

Conclusions and Future Work

This paper has presented the design and implementation of the CDR application, a system designed to support cooperative document management and document processing in a ministerial or governmental setting. It facilitates group cooperation and communication by integrating existing off-the-shelf components into a single comprehensive user interface which allows easy control of the various functionalities available in the different components, which were previously not integrated and therefore complex to employ in a productive manner. The adherence to the metaphor of document-based collaboration as a means of interacting with the system and performing the cooperative tasks has been implemented after an analysis of the collaborative tasks within the context of distributed governmental agencies.

Even though the design and development of the CDR was strongly driven by its current application context of distributed governmental agencies, we believe the system is flexible and powerful enough to be usable in a wide range of application domains even beyond the public administration sector. Its approach of supporting the cooperation and communication tasks involved in joint document creation and document-based meeting support as well as its foundation on commercially available system components make it usable in a large number of distributed settings, e.g. in virtual organizations or distributed teams.

The CDR has been installed in the user community and we are currently gathering feedback from the users which will directly influence the further development of the collaboration solution. Possible extensions of the system include enhanced document management functionalities, e.g. the support of different document versions within the repository as well as advanced document indexing and retrieval features. An extension of the architecture to include several distributed database servers might become necessary in order to better support the distributed setting and address scalability issues. Other aspects that will be examined in the ministerial application context is the use of secure audio/video conferencing, e.g. by using video encryption, as well as an advanced authorization concept for the team room setting to enable the use of the CDR application suite also in security-sensitive environments.

Acknowledgements

The authors would like to thank the following people for the effort they put into the development of the software described in this paper: Marc Volz, Bernd Bussek, Oguzhan Karaduman, Fritz Lorenz, Sascha Lüdecke, Michael Pauly. We are also grateful to Jörg Haake for valuable comments about preliminary versions of this paper.

References

- [1] Engel, A., Kaack, H., Kaiser, S.: Teamarbeitsräume zur Unterstützung verhandlungsorientierter Telekooperation; in: *Rechnergestützte Kooperation in Verwaltungen und großen Unternehmen*, Tagungsband zum Workshop der GI-Fachgruppe FG 551 und der GI-Fachbereiche FB6 und FB8 im Rahmen der Jahrestagung der Gesellschaft für Informatik; Aachen, 1997
- [2] ITU-T Recommendation H.320 (03/96) - Narrow-band visual telephone systems and terminal equipment; available electronically from http://www.itu.ch/itudoc/itu-t/rec/h/h320_23397.html; 1996
- [3] ITU-T Recommendation T.120 (07/96) - Data protocols for multimedia conferencing; available electronically from http://www.itu.int/itudoc/itu-t/rec/t/t120_35511.html; 1996
- [4] Microsoft ODBC 3.0 Software Development Kit and Programmer's Reference; Microsoft Press, 1997

- [5] Knopik, T.; Tietze, D.; Volz, M.; Paul, B.; Heite, R.; Speichermann, H.; Wittinger, C.: Towards a Collaborative Document Archive for Distributed Governmental Agencies; in: Lehner, Dustdar (eds): *Telekooperation in Unternehmen*; Dt. Univ.-Verlag, Wiesbaden, Gabler, 1997; pp. 65-78
- [6] Graham Hamilton, R. G. G. Cattell, Maydene Fisher: *JDBC Database Access With Java: A Tutorial and Annotated Reference (Java Series)*; Addison-Wesley, 1997
- [7] Streitz, N.A., Geißler, J., Haake, J.M., and Hol, J. DOLPHIN: Integrated Meeting Support across LiveBoards, Local and Remote Desktop Environments. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94)*, Chapel Hill, N.C., October 22-26, 1994, pp. 345-358.
- [8] Appelt, W.; Busbach, U.: The BSCW System: A WWW-based Application to Support Cooperation of Distributed Groups. In: IEEE (ed.): *Proc. of the Fifth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, June 19-21, 1996, Stanford, CA. Los Alamitos, CA: IEEE Computer Society Press, 1996, p. 304-309.
- [9] project description available at <http://orgwis.gmd.de/projects/COOPWWW/>
- [10] Information about CUSeeMe is available electronically from <http://bio444.beaumont.plattsburgh.edu/CUSeeMe.html>.
- [11] Information available electronically from Lotus Realtime Notes Web site at <http://www.lotus.com/home.nsf/welcome/realtime>
- [12] Roseman, M., Greenberg S.: TeamRooms: Network Places for Collaboration; in: *Proceedings of CSCW'96 (ACM Conference on Computer-Supported Cooperative Work)*; Cambridge, MA, USA; 1996