Proceedings of the Third European Conference on Computer-Supported Cooperative Work
13-17 September, 1993, Milan, Italy
G. De Michelis, C Simone and K. Schmidt (Editors)

TOSCA Providing organisational information to CSCW applications

Wolfgang Prinz
GMD - German National Research Centre for Computer Science, Germany

Abstract: Most cooperation support systems require information about the organisational context in which they are used. This is particularly required when systems are used in a large organisation or for the support of inter-organisational cooperation.

Following from this requirement, this paper presents the design and functionality of the organisational information system TOSCA for cooperation support systems. TOSCA is composed of two major components: an organisational information base server, which provides services to applications and an organisational information browser, which provides user access.

The paper describes the motivation for an organisational information system, the object oriented data model that is used for the information representation, the architecture of the overall system, and the design of the user interface that presents and provides access to the multimedia information. It concludes with the description of how this system supports a task management system and the role it would play in a CSCW environment.

1. Introduction & Motivation

The overall aim of TOSCA[1] is the representation of knowledge about organisations and their resources which are relevant for the support of communication and cooperation. Major issues of the system are:

- **provision of organisational context information**

Cooperation in teams and organisations is always embedded in an organisational framework. This requires the provision of information about the organisational context in which users work which helps to choose the right patterns for

[1] The organisational information system for CSCW applications

The work described in this paper has been partly supported by the CoTech project MOCCA and the ESPRIT Basic research action COMIC.

communication and cooperation. Information must be provided to answer questions such as: Who is responsible for carrying out a specific task? Whom can I ask for help? Furthermore the system should provide information as to how particular tasks are handled in the organisation. What are the organisational rules one has to consider? Whom do I have to ask first? Which document type do I have to use? All this information belongs to the knowledge which is normally not or only very implicitly provided by CSCW applications, although it plays a significant role in cooperation. TOSCA provides this information to users and applications. Thus, not for every application which requires this information an own information base must be developed and managed.

- **distributed provision of directory information**

Communication requires reachability information about the cooperating partners. This includes communication addresses as well as information about reachability and preferred communication methods. Most applications tackle this by providing simple address directories, which are often not distributed and furthermore can not be shared between different applications. So, in the worst case, each user manages his own directory for each cooperation support application he is using. It is one of the aims of the organisational information system presented here to overcome this problem by providing and integrating a distributed directory service to applications and users.

- **integration of standardised external resources**

An approach chosen in (Hennessy et al., 1993), which we also have investigated (Prinz and Pennelli, 1992), is the application of the X.500 Directory. X.500 is a CCITT and ISO standard for an electronic address book (X.500, 1992). With its potentially world-wide distribution, its methods for distributed management, and its standardised service interface, it fulfils the requirements for a distributed address directory and scalability. However, shortcomings arise when the directory is applied to a more detailed modelling and administration of organisational information. Major problems deal with the representation and modelling of organisational relationships and data integrity. Nevertheless, in order to benefit from the existence of X.500, we found it important that TOSCA integrates access to the X.500 world.

- **integration of cooperation resources**

Comprehensive cooperation support benefits when the resources for cooperation support such as documents, calendars, structured message types (Pankoke-Babatz, 1989; Borenstein, 1992; Malone et al., 1992) can be integrated with the context in which they are used. The advantage of TOSCA is that it is more than a storage server for this information. It allows the association of this information to its organisational context, i.e. by linking it to the projects, departments, etc. where they are used or to the people who use it.

- **support for scalability**

Organisational information is of particular importance in large geographically distributed organisations and for the support of inter-organisational cooperation (Engelbart, 1990). This raises the aspect of scalability which we see as a crucial issue for the success of CSCW applications. From the administrative viewpoint it must be easily possible to extend the number of users of an application. This requires an underlying distributed service environment which provides a set of common services needed by cooperation support service. The organisational information service presented here is one fundamental component in such an environment. As a support service it simplifies the introduction and use of new applications into the working environment and this may increase the acceptance of these services (Markus and Connolly, 1990).

- **visible and user tailorable model**

We aimed to develop a flexible data model that allows an adaptation of our system to various organisations. This is required, because organisations change and it is impossible to develop a single representation that fits all considerable organisations. For that reason an object oriented approach has been chosen. Together with the provision of an object modelling tool TOSCA provides visibility of the concepts and allows users and groups to tailor the object model to their specific need.

This paper is organised as follows. First we will present the data model for the representation of organisational information. The design of an organisational browser is described afterwards, showing three different scenarios of use. Then, the architecture of the system is presented followed by an examination how the system is integrated into a larger CSCW environment. The paper concludes with a brief description of future plans and a summary.

2. A Data Model for the representation of organisational information

2.1. General considerations

Before we explain our data model we should define what we understand by organisational information. We consider organisational information to be information about the entities of an organisation that determine and describe the working context of users. This includes information about the employees, projects, roles, committees, departments, locations, etc. of an organisation. Furthermore, the resources of cooperation such as documents, calendars and other kinds of commonly used data must be considered. In order to provide helpful information on how to perform tasks in an organisation, the system needs to represent guidelines which can be used as resources for planning and carrying out a cooperative activity.

All these discrete bits of information become expressive only when they can be related to each other. Therefore we need ways to describe organisational relationships such as: who is member or leader of a project, which projects are undertaken by a particular department, who is the projects secretary, who is occupying the role of the technical administrator of a special file-server, or who supports which task, or which forms do I need to apply for an organisational procedure? It is also necessary that these relationships can be defined in a dynamic way according to the organisational rules. For example, if a committee consists of the members of the projects of a department, we do not want to list all these people explicitly as would be required in X.500, but we want to express this by an appropriate rule. This reduces redundancy, management overhead and increases consistency when the information is changed.

These requirements and the fact that we wanted to develop a system that is extendible and tailorable to different organisational settings led to the decision to choose an object oriented model for the representation of the organisational information.

The meta object model distinguishes between organisational components and relationships between these entities. Two different basic object types are defined for the representation of organisational objects and organisational relationships. Based on these basic object types a comprehensive set of subtypes for the specific representation of organisational information is defined. The definition of a type requires the specification of several properties each instance of the type must fulfil.

All organisational objects and relationships inherit from basic types which implement the required methods to access and manage object instances. Additional methods can be added for subtypes if needed, but this requires programming which is not expected to be done by end users. Therefore we will focus only on the structural and not on the functional issues of the object model.

2.2. Model of an organisational object type

Organisational object types are used to define a schema for the representation of organisational components. Such a type is specified as follows:

- type name & super type
- corresponding X.500 object class name
- scope of type definition
- textual description of the type
- user friendly name construction rule
- graph layout description
- mandatory and optional attributes
- mandatory and optional relationships

The **type name** is a unique identifier which should describe the semantics of the type. The model provides single inheritance, i.e. one **super type** must be identified.

If available, the name of the corresponding **X.500 object class**[2] can be provided This information is used to map X.500 entries which have been retrieved from the Directory onto the appropriate object type in the organisational data model.

Assigning a **scope** to a type restricts its use to a special organisational context, e.g. a project or group. For example, the usage of a type can be restricted to the context of a group which avoid type clashes and a proliferation of types throughout the whole distributed system. However, users must be aware that it might also hinder cooperation.

The name of an object is often not expressive or user friendly enough for its use in a user interface. Therefore a construction rule for a **user friendly name** can be supplied for an object type. This name can be built by a combination of attributes as well as by retrieving information from objects that this object is related to.

To supply a description of the context in which an object is embedded it is very helpful to provide a graphical view. A **graph layout description** is used for the creation of organisational charts which describe the organisational context of a focused object.

The object model distinguishes between **mandatory and optional attributes** of an object type. Attributes can be basic data types but they might also contain picture, audio or video information (see the section on the user interface). Furthermore attributes can contain expressions which are evaluated on access. These expressions are used to refer to other objects, to express general rules, or to generate a value from others.

Relationships between organisational objects are represented by objects of a special relationship object type. Like attributes, relationships can be either mandatory or optional.

In the past, approaches to model organisations were mainly undertaken in the context of organisational science for the analysis of organisations (Heilmann et al., 1988), or for the planning and support of office procedures (Rupietta, 1990; Victor and Sommer, 1991). Although these systems do not address all the specific issues of TOSCA, our object model has been influenced by their investigations in organisation modelling.

We have developed an object model for GMD. During this exercise we found it very useful to be able to change the object model on the fly. This allowed us to react immediately to new requirements which where raised during data acquisition. The same experience was made when the system was used as a demonstrator for other organisational settings. A full description of all object types that have been defined for our prototype system is not possible in the framework of this paper.

[2] The notation X.500 object class corresponds to the notation of an object type in our object model.

2.3. Model of an organisational relationship type

The organisational relation object type has the following characteristics:

- type name & super type
- corresponding X.500 attribute name
- relation identifiers
- value set attributes

Type name & super type are same as for the organisational object type.

If available the name of the corresponding **X.500 attribute type name**, is used to map X.500 attributes which have a distinguished name syntax, i.e. which point to another entry onto the appropriate relation object type and vice versa.

A relation object describes a relationship between two organisational objects. Depending from which entity the relationship is viewed, it needs to be denoted differently. For example a project membership relation between a project and an employee object is called "has members" from the project view, but "is a member of" from the employees view. These identifiers are called **relation identifiers**.

Relationship objects include two **value set attributes** (source and destination) which contain the description of the related organisational objects. These objects can be described by naming them or by expressions which allow for a dynamic description of the relationships. These expressions allow the description of organisational rules such as: voting members of this committee are the project leaders of all projects of the department. Also, they can be used to reduce redundancy by describing rules, such as: employees of this institute are the members of all projects of this institute. Furthermore it is possible to define user dependent rules. This is needed for example, when the person who is responsible for a task, depends on the users membership in a project. In this case the actual user identity is needed to answer a request.

2.4. The object model designer - creating and extending the object model

The meta object model was designed to be tailorable to various organisational settings. For that purpose, a window-based object model designer has been realised. This tool allows administrators to create and modify the object types which are used to represent the basic organisational components such as project, department, etc., i.e. which are needed for the structural modelling. Users are allowed to extend the model by definition or subtyping of types which are relevant for their local or cooperative work. For example these are types for the storage of addresses, or for the representation of shared working resources such as texts or notes.

An interesting application of that functionality is the definition of message type objects similar to the approach of semi-structured messages presented in (Pankoke-Babatz, 1989). Members of a project might define their own message types which are used to exchange meeting dates, automatic generated notes, etc. To support such applications the object model defines some basic message object types which can be subtyped for further purposes. Although this approach is comparable to the one taken by the Oval system (Malone et al., 1992), the difference lies in the fact that our system allows the association of these message types to an organisational context, e.g. a project or a committee. This scoping of object types helps to avoid a proliferation of types throughout the whole system. Supplementary solutions for that problem can be found in (Lee and Malone, 1990; Johnson, 1992).

Although the system has not been primarily designed for that purpose, the facility of a user tailorable object model combined with automatically generated forms, makes the system applicable for a simple emulation of cooperative hypermedia applications (Haake and Wilson, 1992). The additional advantage is that our system allows an association of such documents into their organisational context, e.g. a hypermedia document can be easily linked to the appropriate project in which it was produced. Thus, organisational context information is augmented with working resources, and vice versa the resources are linked to their originating context. This provides access to the information via different associations and from different context.

3. The Organisational Information Browser

3.1. Introduction

The organisational information browser provides user access to organisational information. Three major patterns of cooperative work are supported. First, it allows access to and multimedia presentation of organisational information Second, the interface integrates different communication media to support ad-hoc communication. Third, in combination with a task management system, it provides means for the planning, instantiation and coordination of cooperative tasks.

3.2. Querying and presenting organisational information

Cooperation requires information about the cooperating partners. This ranges from simple address and technical reachability information to their organisational context which helps to choose the right patterns for communication and cooperation. It is furthermore very comfortable, when the resources of communication are integrated and can be accessed in the same way. This section describes how that information is presented by the interface and how it can be accessed.

The interface allows browsing and searching for organisational information and tracing of organisational relations via a graphical window interface. As well as text information the interface is able to present different media which is represented in the information base: graphics used for maps and the presentation of organisational hierarchies, relations, procedures and rules; photos (people, groups, buildings, rooms), audio (explanatory text) and video (video demonstrations of software, presentation of public services, etc.).

Normal user interaction starts with a window that provides browsing and querying functionality, as well as means for an easy switching between both search methods (Figure 1).

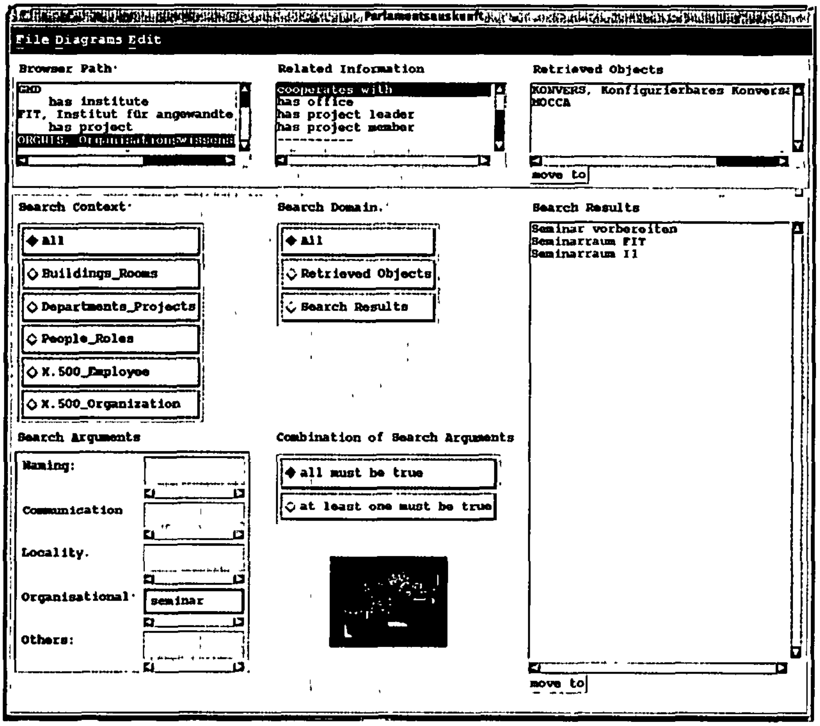

Figure 1 The browser and search window

A set of predefined windows for the display of particular object types and their organisational interrelations has been implemented. In addition, for those which don't have a special presentation a generic window is displayed that is automatically generated from the object's type information. This reacts flexible on model extensions done with the object model designer presented above.

The whole system is realised as a hypermedia interface. Thus, whenever information is displayed which refers to another information object, this can be immediately retrieved by a user action. That allows manifold ways to access and browse through the organisational information, but it also expresses the various relationships which exist.

Using the mapping information provided as type information for each object the user interface is also able to display objects which have been read from X.500 directory. This is useful for example, when an international project description contains references to members which are not stored locally but represented as entries in the X.500 directory. Thus, the administration of that information is done remotely by that person, while we still have access to it. This reduces redundancy and guarantees actuality. This external information object is viewed like an internal one. Of course, the user sees a difference in the richness of the data, because X.500 doesn't provide the same amount of data and relations as our system.

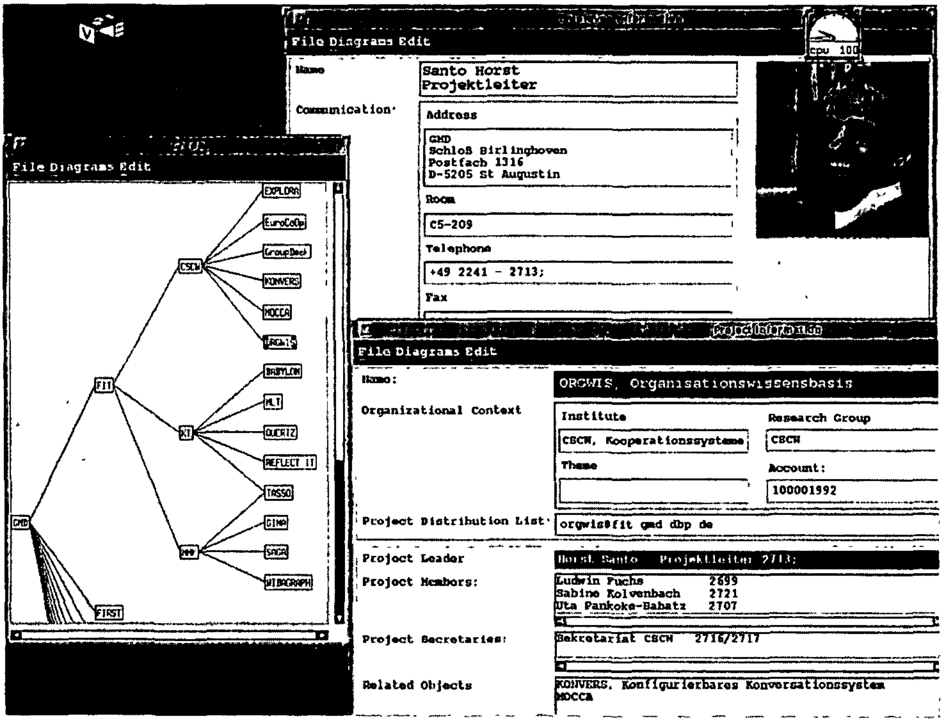

Figure 2 Windows displaying partial information about a project, an employee and a graph that displays the organigram for the project and its supervising institute.

For getting an overview on an organisational object and its relationships a graph can be displayed. This is typically an organigram that shows an object in its organisational context. Figure 2 shows a graph for an institute. The graph shows the research groups and projects of this institute, as well as the other institutes of the organisation. It is generated interpreting the object types graph description. The graph can be used for further browsing, i.e. by selecting any of its entries the appropriate object is displayed.

When pictures or maps are used to represent information, they can be used for browsing, too. Linking a picture object by special relationship objects to other objects, areas of a picture become sensitive, so that additional information, e.g. a more detailed map, or text information is displayed when this area is selected. Audio information can be used to give additional online description.

Cooperation support systems and their user interfaces can not be treated as closed applications. Integration and interworking with other applications must be possible (Engelbart, 1990). In our system this has been achieved via object adapters for external applications or data. By that technique we have integrated calendars, document editing systems, etc. into our organisation browser. This allowed us to turn the information system into a kind of general desktop interface that groups the working resources of a user according to his organisational context (private and project calendars, project papers and documents, etc.).

3.3. Support for ad-hoc communication

For the support of immediate communication with partners or about resources that have been identified it is very important to integrate communication support applications. Currently we have integrated mail and a broad-band video-conferencing tool that has been implemented locally. So, when appropriate, a user can send mail to a person he has just retrieved, or start a video conference with a partner without launching another application first. The system provides information about the communication partners preferences as well as their technical infrastructure. This helps to avoid unsuccessful contact approaches and delays. Obviously, this will not replace the standalone communication applications, but the idea is to provide a stronger integration of these applications also into other applications which deal either with resources that can be communicated or with communication partners.

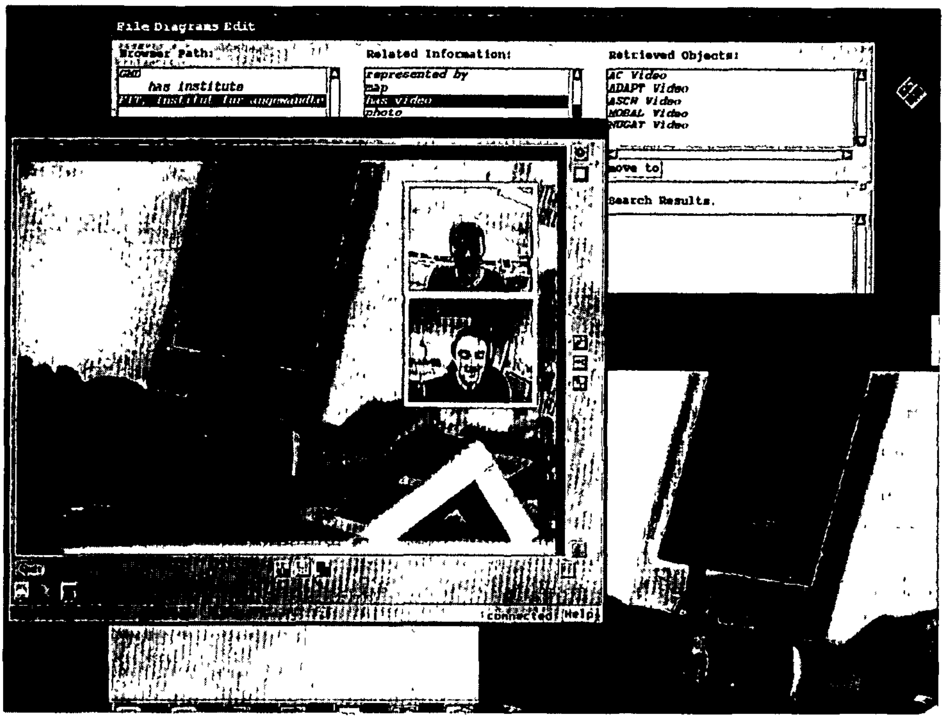

Figure 3. A video conferenced has been launched in which a video retieved from TOSCA is shown to the communication partner.

Figure 3 gives an example for a scenario which illustrates the integration of video-conferencing with the browser's ability to manage and present video clips. A user has started a video-conference using our locally developed video-confemcing system LIVE (Fuchs, 1993), with a partner he retrieved using the browser (upper left window). Then, he selected a project video from a project entry (lower right window), which he is now transmitting in the conference. The video is retrieved from a video tape which contains a set of video clips. It is controlled remotely by our application using time coordinates which are stored for each clip in a video entry. This scenario exemplifies the benefits received by the integration and support of a video-communication tool with the organisational information system.

For each object that is displayed a simple white board functionality is provided. This allows users to communicate on information they have found in the system. For that purpose, comments can be patched on each object (similar to yellow post-it notes). These comments can be viewed, added and modified by all users. It can be used to leave useful experiences or to express problems for other users who lookup the same information. It can also be used as by a group of users who start a discussion about an information object, e.g. about possible extensions on a service that is described, or about informal work-arounds for organisational procedures. With that functionality a communication and discussion tool is directly integrated with the context of discussion, i.e. with the information and its organisational context that caused the discussion.

3.4. Support for cooperative tasks - application by a task management system

TOSCA provides means to describe how tasks or procedures can be carried out in an organisation or in a group. This is represented in task template objects according to a model that has been developed by our local partner project at GMD (Kreifelts et al., 1993). A task is described in an outliner format. This allows the description of major and subordinate tasks. For each single task it can be specified who can support that task or who is responsible to carry it out. Furthermore, resources can be associated to each task, such as documents, forms, calendars, etc. This is done by appropriate relationship objects. These are described user specific. Each user gets individual information about the people who are responsible or the forms which are valid for him. Thus, TOSCA represents abstract templates which are interpreted and individualised on retrieval.

We would like to stress that the task templates are understood as resources for users to develop their own plan. They are not intended as a prescription how a cooperative task must be carried out (Robinson and Bannon, 1991).

Task lists can be interrelated, so that users are informed about alternatives or related templates. This increases the visibility of organisational procedures (Schmidt, 1991). The white board functionality can be used to comment on work-arounds or experiences one has had in carrying out a task.

Although this information is already very useful as a resource to initiate and carry out a cooperative task, it becomes more useful when it can be transferred into a system that supports its coordination. That integration has been realised with the task management system in the framework of the ASCW prototype (Hoschka, 1991; Kreifelts et al., 1993). Users can export a task template from TOSCA and then import it into their personal task list. This is convenient for routine task descriptions and it helps when the user carries out a task for the first time. Then the distributed execution of that task is supported by the task manager. In the further process TOSCA is used for address lookup, to resolve role descriptions when administrative offices are involved, or to look for substitutes, etc.

4. Architecture & Implementation

TOSCA consists of two major components, an organisational information server and the organisation information browser. The server stores and manages the organisational information objects and relations.

The server is realised on top of a distributed object oriented database (ONTOS). Access to the X.500 world is provided by an integrated directory user agent. All requests for external information are forwarded to the X.500 service. References from the organisational information to X.500 information are automatically resolved. Entries retrieved from X.500 are translated into the internal object schema.

The server provides a data and a schema management application interface to applications. These interfaces are used by the organisational information browser, the object model designer, by communication and cooperation support services.

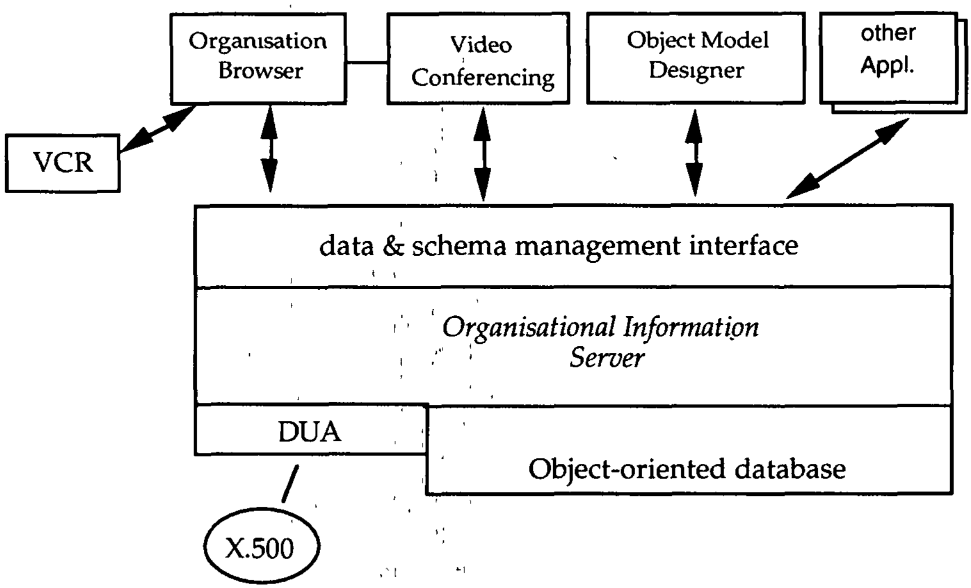

Figure 4.Architecture of TOSCA

The whole system has been implemented in C++, using GINA (Spenke and Beilken, 1991) as an interface toolkit, and Quipu (Kille, 1989) for the X.500 directory components. The server contains currently appr. 750 organisational objects and 600 organisational relationships which are used to represent parts of GMD.

5. Integration with a CSCW environment

With TOSCA, we have developed a system for the support of CSCW applications. However, a comprehensive support requires the provision of additional underlying services among which an organisational information service plays an important role. This is illustrated by figure 5 which outlines the relationships between the developments from e-mail services to CSCW systems and the required supportive services starting from a directory service and ending with a CSCW environment in a simplified way.

As a member of CO-TECH[3] project Mocca (Navarro et al., 1993) we are working on the requirements and design of a CSCW environment. That environment aims to provide a platform for the support, integration and interworking of CSCW applications. Five views on cooperative work have been identified: information, organisation, workspace, distributed architecture, and a rooms metaphor. For the first three views models have been developed while the others lead to the design of an architecture and a virtual world. The organisational model presented in this paper has been chosen as the model for the organisational view.

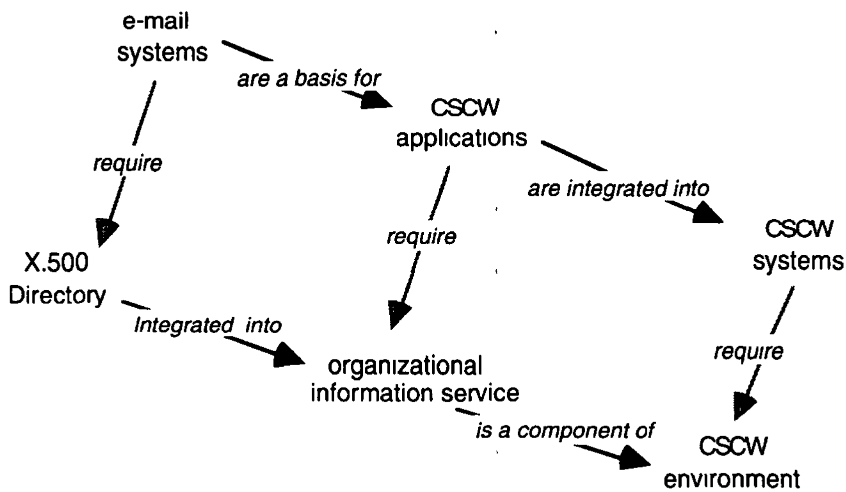

Fig. 5: Relationships between cooperation support applications and supportive systems

Integrated into a CSCW environment, the organisational information service provides a set of services to the other components. To list the major ones:

- Distributed environments require a unique naming and addressing scheme that allows the identification of objects. This naming scheme is provided by the organisational information base for the other environment components.
- Workspaces are used to model shared working areas for groups of people. The definition of a workspace includes the specification of its relation to the organisational context in which the work takes places, e.g. the members of the

[3] CO-TECH is a basic research action aimed at establishing a Europe wide CSCW community.

workspace, a project for which the workspace is created, etc. Since all that information is already present in the organisational information server, no additional information needs to be provided and furthermore access rights or other organisational rules can be automatically derived from the policies described for the members context.

- The virtual world user interface requires an underlying system that provides information about the topology of rooms, access and constraints on rooms, the tools and resources available in rooms, etc. This information is represented in TOSCA. It comprises both, the representation of real rooms, buildings and locations and of virtual ones.

6. Open aspects and future work

Our first prototype will be extended by a more powerful user interface for the distributed administration of the organisational information. Our concept for future developments will allow a distributed administration of the information by users and administrators, since it would be wrong to assume that the whole information base is administered by a single organisation expert. This will be based on the description of access rights and responsibilities. Using these rules, objects will automatically request update-information from the appropriate administrator when they are newly created or when inconsistencies are detected.

Another interesting aspect is the provision of awareness about changes in the organisation. This will be realised using an eventing concept. Modification of objects which reflect changes in the organisation will produce events of different types. Based on the event type an event is then forwarded along appropriate organisational relationships to other objects which then can react on this. For example: The creation of a new employee object in a department produces an event that is forwarded as an informal notification mail to all other employees related to that department while it will also produce a request for the e-mail administrator to install a new account. Or, changing a task template description leads to a notification of all people who are involved.

This work will be partly carried out in the framework of the Esprit Basic Research project COMIC[4]. In the long term, our aim is to extend the organisational information service to a general CSCW object service in the context of the Polikom programme (Hoschka et al., 1993).

7. Summary

This paper presents the motivation, design and realisation of an organisational information service for the support of CSCW applications. We believe that such a service is a fundamental service in a CSCW environment which provides common

---

[4] Computer-based Mechanisms of Interaction in Cooperative work.

services to other applications and serves as a helpful information service for users in their cooperative work. To summarise:

- the system allows the representation of the organisational context for the support of:
  - applications for cooperation support
  - users as an information and cooperation support service
- it provides and integrates different communication media and facilitates ad hoc communication
- it represents task descriptions as resources for cooperation support of users and applications
- it provides a visible and user tailorable object model and thus allows an adaptation to various organisational settings.
- it provides an integrated access to the internationally standardised X.500 directory service
- in order to provide the best possible representation of information, the system is capable of handling multimedia data

The application domain for such a service are larger organisations as well as the support for inter-organisational cooperation which becomes more relevant for CSCW in the future. That requires scalable systems on the underlying support and application level. Our design decisions to realise a tailorable, flexible object model, to integrate X.500 access, and to base our implementation on a distributed system makes the system scalable to a large extend. In addition its use by other applications will also simplify their scalability and technical integration into an organisational setting.

8. Acknowledgements

The system presented in this paper has been realised within the Orgwis project at GMD. Many thanks to all its members for their ideas and implementation work. Special thanks to M. Busch, J. Mariani, U. Pankoke-Babatz, and H. Santo for their comments on this paper.

9. References

Borenstein, Nathanial. 1992. Computational Mail as Network Infrastructure for Computer-Supported Cooperative Work. In *CSCW '92, Toronto, Canada*, 67-74. ACM.

Engelbart, Douglas. 1990. Knowledge-Domain interoperability and an open hyperdocument system. In *CSCW'90, Los Angeles, USA*, 143-156. ACM.

Fuchs, Ludwin. 1993. *LIVE - Videokonferenz auf der Sun SPARCstation*. GMD, Februar 1993.

Haake, Jörg and Brian Wilson. 1992. Supporting Collaborative Writing of Hyperdocuments in SEPIA. In *CSCW '92, Toronto, Canada*, 138-146. ACM.

Heilmann, Heidi, W. Sach, and M. Simon. 1988. Organisationsdatenbank und Organisationsinformationssystem. *Handbuch der Modernen Datenverarbeitung* : 119-129.

Hennessy, Pippa, Thomas Kreifelts, and Ute Ehrlich. 1993. Distributed work management: Activity coordination in the EuroCoOp project. *Computer Communications* 15 (8): 477-488.