

MOCCA - AN ENVIRONMENT FOR CSCW APPLICATIONS

Steve Benford¹, John Mariani², Leandro Navarro³, Wolfgang Prinz⁴, Tom Rodden²

Our belief is that future, Computer Supported Cooperative Work will by necessity involve a heterogeneous collection of applications, paradigms and models and that no single system will meet all the requirements of all groups. For this reason we propose a CSCW Environment, a form of operating system, which facilitates inter-working between many different CSCW applications. We describe the key requirements of such an environment, followed by a set of models which offer different perspectives on the environment and which collectively define its functionality. These models address issues including setting planned work against a background of ad-hoc interaction; providing a common organisational context for applications; defining the structure of shared information between applications; and representing the structure of the work taking place. We conclude by discussing how these models might be used to specify the components of a distributed architecture which implements the environment in an open systems manner. The work described in this paper has been carried out by the MOCCA group (working group 2) of the European CO-TECH programme. CO-TECH is aimed at conducting basic research into CSCW and at establishing a Europe-wide CSCW community through a number of different working groups.

1. Introduction

The label CSCW applies to a wide variety of applications including shared editors, audio/video-conferencing, computerised meeting rooms, group design tools, co-authoring systems, shared calendars, work-flow systems, voting tools, whiteboards and message-based conferencing. Many researchers have attempted to find a common underlying model which could be used to describe such applications and, ultimately, to construct a range of new ones [WINOGRAD 86, BOWERS 88, PANKOKE 89, BIGNOLI 91, KREIFELTS 92]. These models frequently provide a notation or language for “configuring” applications which can be interpreted by a general purpose underlying CSCW system.

The goal of a CSCW system which supports a wide variety of cooperative uses is a worthy one. However, the idea that all CSCW applications can be built to a common model may prove to be impractical. CSCW applications provide diverse models and mechanisms aimed at supporting either a particular cooperative activity or a class of activities. It is likely that this multiplicity of approaches will persist and that users will exploit a range of different applications. Unfortunately cooperative applications are often unaware of the existence of other applications and provide few mechanisms for working in conjunction with other applications. This observation provides the motivation for the work presented in this paper. The goal is to make it possible to run many CSCW applications in an integrated way. However, instead of looking for a monolithic solution to CSCW, we assume that the applications may be based on quite different underlying paradigms or models. Our approach is to develop a **CSCW Environment** - a distributed system that supports inter-working between heterogeneous CSCW applications. The environment is analogous to a CSCW

¹ Department of Computer Science, University of Nottingham, UK.

² Department of Computing, University of Lancaster, UK

³ ETSE Telecomunicacio, U.P.C., Barcelona, Spain

⁴ GMD, St. Augustin, Germany.

operating system and provides a set of management services that allow people to locate and combine heterogeneous tools.

This distinction between a generic CSCW system based on a single underlying model of communication and a CSCW environment which provides support for a range of models is important. Our work focuses on support services for CSCW including information sharing between applications; describing common organisational context and resources; merging applications with background social interaction; navigation and location; and awareness of on-going activity. It is not directly concerned with specific communication processes or application specific functions.

This paper outlines such a CSCW environment, describing its functionality from a number of different perspectives. This work has been carried out in the MOCCA⁵ working group of the European CO-TECH⁶ programme.

2. Goals of the MOCCA Environment

Cooperative work does not occur in isolation. Instead, the general picture is of many inter-related activities situated within a common setting of shared resources, people and information. These activities may vary tremendously in terms of communication structure, the flexibility of the tasks involved, timescales and the numbers of participants. Unfortunately, the current generation of CSCW applications often provides models and mechanisms aimed at supporting a particular philosophy or approach to cooperation. Thus, users of a particular CSCW application are presented with a specific interpretation of cooperative work and can only operate within the confines of that closed world.

The reality of supporting cooperative work is that a wide range of CSCW applications, each exhibiting a distinctive model of cooperation, need to work in unison. Consequentially, the role of the computational platform in which CSCW applications exist becomes a crucial factor for the future success of CSCW. The central aim of a CSCW environment is therefore to provide mechanisms to ensure that applications can work in harmony rather than in isolation from each other thus allowing a multiplicity of CSCW approaches and paradigms to co-exist. The Environment aims to manage people, information and other resources and to share them between a set of applications. At an architectural level, these functions would be provided through a set of common service interfaces which applications can use to establish a working context. The general goals outlined above have been used to generate a more specific set of requirements for a CSCW environment which have driven the work of the group so far. These include:

- *Information sharing* - the environment should allow different CSCW applications to share and exchange information.
- *Representing Organizational context* - the environment should manage knowledge about organizations and their resources and so provide a common context for applications.
- *Activity relations* - the environment should be capable of representing relationships between different applications (e.g. the temporal relationship "schedule application A for when application B completes").

⁵ MOCCA is one project in the CEC's COST 14 program. The acronym could refer to "Models for Open Comprehensive/Cooperative CSCW Applications". In reality, this is just a weak justification for an acronym and its real interpretation is lost in time.

⁶ CO-TECH is a basic research action aimed at establishing a Europe wide CSCW community. Funding is provided for travel and exchange of researchers with the aim of generating ideas and directions to be picked up in future projects.

- *Transparency* - to simplify interaction, hiding some dimensions that are unnecessary for the cooperative activity and that make the system look more complex. This mechanism should be selectable rather than imposed.
- *An Open Approach* - the environment should operate in an “Open ” manner. This requires integration with international standards for distributed systems - either *Open Systems Inter-connection (OSI)*, or more latterly, *Open Distributed Processing (ODP)*.

It is necessary to stress the importance of “scale” for the MOCCA environment. MOCCA is concerned with interaction on a potentially large scale. This implies support for many users spread across a large number of organizations. In such an environment, issues such as location, navigation, organizational context and standardization assume an importance that is not always so evident in small-scale stand-alone applications.

3. The MOCCA Approach

A CSCW environment represents an extremely complex system and is difficult to consider as a whole. Consequentially, MOCCA has identified a number of perspectives from which to view the world, where each perspective offers an abstract view of specific environment functionality. More formally, the perspectives can be refined to a set of “models” which collectively specify the environment. Adopting an object-oriented paradigm, the environment consists of a set of objects representing people, applications, organizations and other resources. Each object may appear in several models, being viewed differently in each. It should be stressed that the process of decomposition of models within MOCCA involves by necessity a diverse range of models to exists. Two major classes of model are used, *prescriptive models* which state the properties of a model and how these properties may be realised and *descriptive models* which describe what a model should contain but are neutral about how this is realised. Figure 1 shows the viewpoints which lead to the models that currently define the MOCCA Environment.

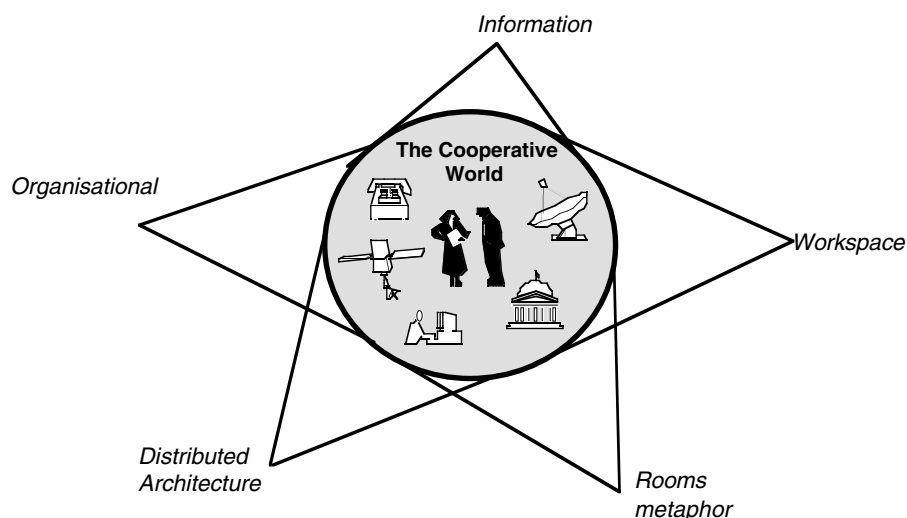


Figure 1: *The MOCCA Viewpoints on CSCW*

The *Rooms metaphor* provides a users conceptual map of the environment and addresses the issues of location, navigation and social interaction in a distributed system. The *Organizational Viewpoint* considers the organizational context required for CSCW in terms of a set of resources. The *Information Viewpoint* examines the definition of shared information, relationships between information objects and the

naming of information. The *Workspace Viewpoint* groups together the various components needed to support cooperative work. The *Distributed Architecture* considers how the models can be used to specify components of an open, distributed CSCW platform. All of these viewpoints need to be influenced by a *Social Perspective* to encourage this MOCCA aims to develop “policy free” models where possible. This means models which recognise the importance of social processes in interaction and which do not prevent such processes taking place (e.g. by being too prescriptive or inflexible). This social perspective is not a computational model in its own right. Rather, it is a driving force behind the other models.

4 The Rooms Model: *supporting the informal basis of work*

As part of the MOCCA project, a metaphor based on the concept of a virtual world which can be explored and inhabited by users is being exploited to highlight the issues surrounding the MOCCA environment. This metaphor is called the *Rooms model* and its goals are to provide a user conceptual map of the environment and to provide a readily available mechanism to investigate the role of each of the other viewpoints. We have found that this metaphorical approach helps clarify the environment in our own minds and we hope that it will play an equally useful role in communicating our ideas to others. The danger is that the use of a “real-world” metaphor like this might limit people’s view of the environment unnecessarily. It should therefore be stressed that the Rooms model defines one of many possible representations of the environment. In addition to this general role, the rooms model is specifically responsible for a number of key issues including supporting for mapping and navigation in a global virtual world; providing a backdrop of informal and ad-hoc interaction against which more planned activity takes place; and encouraging peripheral awareness of other people, objects and events in the virtual-world.

A Room is a virtual space where people interact with each other and with various tools. The use of virtual rooms as a basis for user-interfaces or for “social browsing” has been incorporated in several systems including Rooms [CLARKSON 91], VRooms [BORNING 91] and CRUISER [ROOT 88]. In addition to exploiting rooms as a conceptual mechanism for describing and characterising cooperative work MOCCA proposes several extensions to the basic metaphor. First is the idea of a continually available “ether” allowing informal communication to take place in a room at all times. Thus, a room provides open, continuous communication channels between all its occupants allowing ad-hoc interaction. The importance of focusing on informal communication within CSCW has been stressed by several researchers including Gale [GALE 91], Root [ROOT 88] and Gaver [GAVER 91]. The basis of the MOCCA Rooms model is that *informal communication is the starting point for CSCW* and should always be possible. A second extension is to facilitate “awareness” of who is present, who enters and who leaves a users working environment.

5. The Information Model: *representing shared information*

The Information Model defines mechanisms for representing the structure of information in the CSCW environment and for describing how it is shared between different people and applications. The model focuses on the information that is directly produced, shared and consumed during cooperative work (e.g. messages, documents and conversations) as distinct from information which represents organisations. The Information Model, the Organizational Model and data models such as the *Relational* and *Object-Oriented* models are closely related. Traditional data models define the

underlying modelling technique for describing both the Information and Organization models. Within MOCCA the Object-Oriented paradigm provides the basis for our data model, not least for its widespread use in distributed systems modelling. Figure 2 shows the divisions between the data-model which specifies our modelling approach and the Information and Organization models which define functionality in the MOCCA environment. It also highlights the distinction between the Information Model and the Organization model in terms of the kinds of objects they consider.

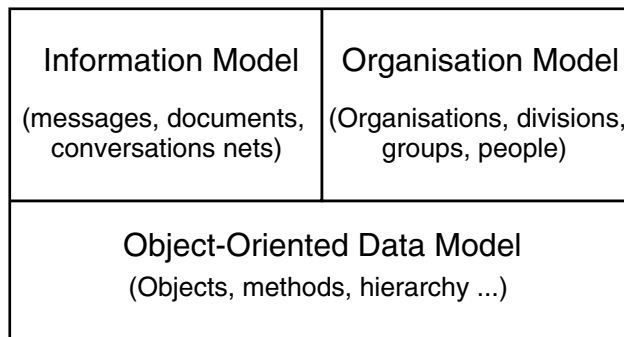


Figure 2: *Relationship between the Information, Organization and Data models*

The main goal of the Information Model is to allow different applications to define the structure of their information in a common format in order to exchange and re-use it. Several CSCW projects have adopted an information sharing paradigm for group interaction (in contrast to the more procedural view of coordinated exchange of messages). These include the AMIGO MHS+ project [SMITH 89] and more recently the GRACE project [BENFORD 91, SMITH 90]. These projects have been strongly influenced in turn by experiences with large scale bulletin board and message-based conferencing systems [HORTON 87, HILTZ 85, PALME 90]. An international standard Information Model is currently being defined as part of the definition of standard support for OSI group communication based on the X.400 Message Handling Service [ISO-GC 91, BENFORD 92]. Given the body of existing work on Information Modelling for CSCW, we have decided not to redefine a model from scratch. Instead, the goal of our work has been to examine current approaches in order to determine the needs of a CSCW environment.

The MOCCA work has currently identified a number of important areas which previous work and emerging standards have not yet addressed. These include transaction support so that CSCW applications can define “long transactions” which access many information objects; schema definition mechanism; specifying an access control mechanism with supports a range of locking policies and mechanisms for defining different views of an object.

6. The Organization Model: *representing the organisational context of work*

Cooperation in teams and organizations is embedded in an organizational framework. For example: cooperating partners are embedded in larger organizations, departments, internal or external projects, they have superiors, colleagues and substitutes. Furthermore, cooperation requires information about “reachability” (e.g. postal address, phone numbers, fax numbers and e-mail addresses). Systems for cooperation support can not assume that the cooperating group is already formed per se. They should support the identification of competent partners by answering questions like: Who is responsible for a particular task in my organization? Whom can I ask for help? Furthermore, the system should provide information about how particular tasks are handled in the organization. What are the organizational rules one has to consider? Whom ought I to ask first? Which document type do I have to use?

These, and more, belong to the implicit knowledge which is normally not available in a systematic form, although it plays a significant role in cooperation. Such knowledge is particularly crucial for large geographically distributed organizations.

The aim of the organizational model is to provide appropriate means for the specification and management of this information. An object oriented approach has been chosen for the modelling of organizational information. Basic objects are defined for the representation of objects and relationships. These objects are then refined to satisfy the specific needs of a particular organization.

This object library provides access to organizational information as well as transparent access to the X.500 Directory service. All requests for information which does not belong to the organization is forwarded to the X.500 service. This avoids multiple databases and ensures consistency as long as the X.500 information is up to date. The relation between the X.500 Directory and the organizational knowledge base can be seen as follows. The Directory provides means for a global provision of public communication relevant information. The organizational knowledge base is used for a detailed representation of organization specific information. The following diagram display the requirements and relations between theses supportive services and CSCW applications.

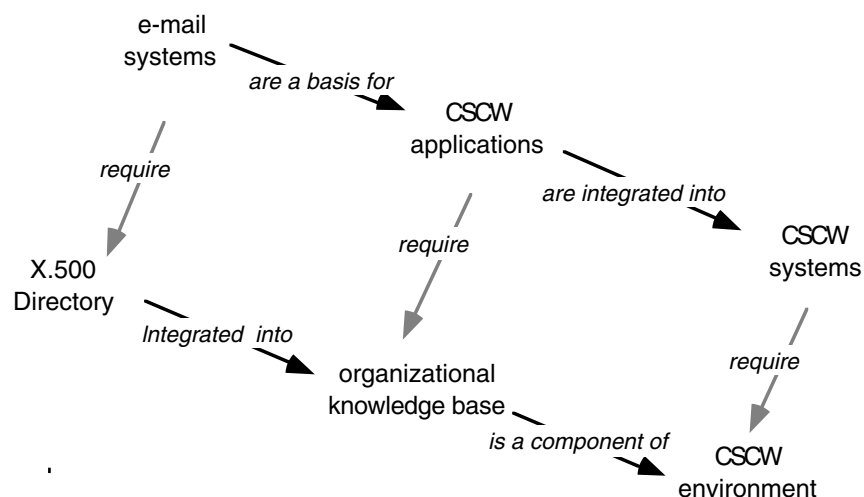


Figure 3: Relationships between CSCW applications and supportive services

7. The Workspace Model: representing the work taking place

A large number of cooperative applications focus on the representation and management of work as a set of related cooperative activities. Different models of cooperative work have been developed within CSCW and a number of these are realised within a variety of systems. Rather than consider a general means of describing cooperative work which is suitable for all situations the approach of the MOCCA environment is develop a set of mechanisms which allow the appropriate features of cooperative work to be represented within the environment. The interpretation and semantics of these representations are left to the application.

Our starting point for considering cooperative work is a consideration of the compositional nature of group work and the development of mechanisms to representation of this composition. The basic unit of structure within the environment is the workspace. A workspace represents the people that belong to the same working group, the objects they use to effect their work and the composition of the work. A workspace is represented with an object of the form:

```
class workspace is
    Members;
    Objects;
    Naming_context;
    Event_templates
```

Where

- Members** - The members of the workspace.
- Objects** - The objects of interest to the workspace
- Naming_Context** - The naming context associated with the collection of objects.
- Event-templates** - The events of interest to the workspace

MEMBERS

Workspace members may be either users or roles. Support for roles tends to be domain specific and semantically loaded. It is up to particular applications manage and enforce the use of roles. However, the co-location of users and roles are represented within a workspaces and workspaces provide a means of registering this relationship dynamically within the environment.

EVENT TEMPLATES

Event relationships are the relationship between workspaces defined within the supporting infrastructure and the events of interest to those workspaces. For example, a student taken course CS361 who is a member of a particular workspace would be interested in all events concerning that course. This interested in events of associated with course CS361 is recorded within the workspace using an event templates.

Event templates are used to compare instances of event objects to decide if they are of interest to members of the workspace. Events of interest are automatically routed to the members of the workspace. Each event template is similar to event object instances with comparison values rather than attributes. This method of comparison is similar to the approach used within the information lens [Malone 87]. An activity template which successfully matches events for course CS361 would be of the form:-

```
sender: = course_CS361
destination:
event_type:
contents: .
```

A similar activity template which considered only the lecture events from course CS361 would be of the form

```
sender: = course_CS361
destination:
event_type: = lecture_event
contents: .
```

When an event matches a template members of the workspace are notified and can access a copy of the associated event_object.

NAMING CONTEXT

Workspace have an associated naming context which maintains local naming conventions for objects within the workspace. This allows locally held object names to

be resolved to a global object identifier. Local names can be used by applications to access these objects.

OBJECTS

Workspaces also collect together the objects of interest to a group. This is achieved by collecting a set of object adaptors [TREVOR 93] within the workspace definitions. Object adaptors represent and control the access to objects and may present a different set of operations (an interface) to users within a workspace. Thus an object is sensitive to the context from which it is accessed. However the basic object itself may be unaware, but is presented in different ways to users depending upon the workspace. The object may present a slightly different interface to reflect the invoking user's rights. The problem with this variety, is that each requires a different form of access control.

Accessing an object from inside of a workspace means that the object may respond differently depending on the accessing user. Every object has a basic set of operations. When an object is created by a workspace that workspace is given an object adapter that abstracts the basic object into a set of workspace meaningful operations, together with a set of rules for that activity. The object adapter itself is defined as part of the workspace in the first place and is stored, along with pre-defined templates (such as exam workspace, conference workspace etc.) in the information store.

WORKSPACE RELATIONSHIPS

Workspaces have composition relationships which link them. These composition relationships allow the composition of working groups to be represented within the environment. These relationships are used to link workspaces. A wide range of relationships can exist within the environment however two relationships are of interest "sub space of" and "related to" which provide links between workspaces. The sub space of relationship allows the traditional hierarchical decomposition of working groups into sub groups to be supported. For example the workspace "CS 361 course" may have the sub_space relationships shown in figure 4.

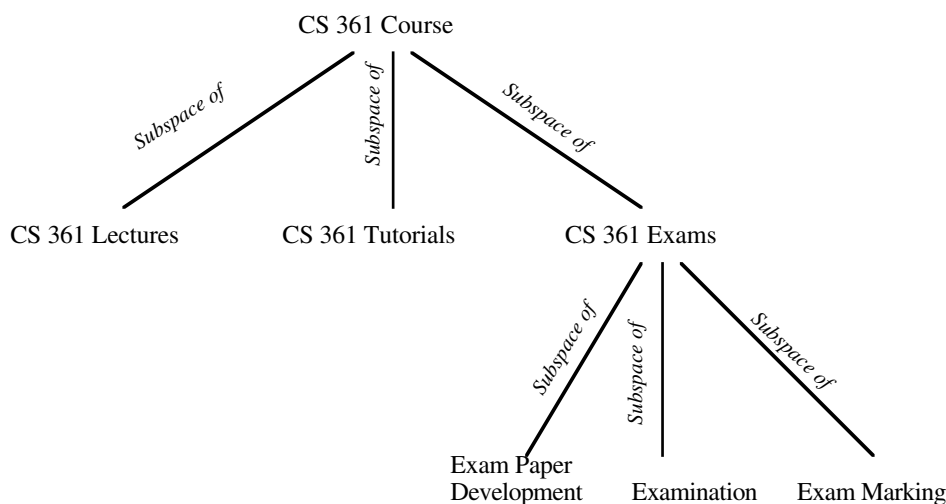


Figure 4: Representing the decomposition of work

7.1 Active workspaces

As we have seen the basic unit of structure within the mocca environment is the workspace. Workspaces allow the grouping of people and information to be represented and managed within the environment. Actions within a workspace are

made visible to others by events. This also provides the basic components for representing activities in the MOCCA environment.

A MOCCA active workspace is a special form of a workspace which provides the means for the current state of an associated on-going activity to be shared within the environment. The compositional facilities provided by the workspace allow the appropriate people and resources to be assembled and managed within the environment. These facilities are augmented by the inclusion of state information to form the active workspace object.

class active-workspace subclass of workspace
Active Status
Activity Templates

ACTIVE STATUS

Active workspaces are intended to support the externalisation of activities and the effects of applications within the mocca environment. In this light as far as a MOCCA environment is concerned an activity consists of a collection of people and objects with an associated *externalised state* representing the status of the particular activity. Alteration of the status of an active workspace is carried out by the applications and users driving the workspace and not the mocca environment. This separation ensures that the MOCCA environment provides only the mechanisms to allow CSCW applications to coexist while issues of policy are left purely to the application.

Active workspace class information

The definition of workspaces is simplified using class templates. Each instance of a workspace is a member of a known workspace class within the mocca environment. Each class definition enumerates the possible states that an activity can have for example a *lecture_course* workspace could have the following activity status values

workspace_status lecture_course is (registration, lecture(1..4)⁷, exam, results_posted)

Active workspace instance information

Each active workspace is registered in the mocca environment as an object and is an instance of that class. Each of these active workspace objects have the following attributes

Workspace I.D - A unique identifier for the activity object

Workspace Class - The activity class.

Plus attribute values associated with the current status of the active object. These instance variables apply only to the specific activity object instances. Both the I.D. and the Activity Class are fixed. The status can only be externally inspected but cannot be amended. The status value can only be amended by members of the workspace.

ACTIVITY TEMPLATES

Another feature of most activity models is the dependencies between activities. For example, many activity models allow users to record that an activity A is dependant on

⁷ Brackets are used here to show numerical states. Lecture(1..4) means we can have the states lecture(1), lecture(2), lecture(3), lecture(4).

B finishing. Thus we need to allow a relationship between one activity and the state of a second activity to be made public. In MOCCA this dependency of action is represented independantly of activity decomposition which is modelled by relationships between workspace. In active workspaces this dependency is recorded by using activity templates. As in the case of event_templates each activity_template is used as a means of comparison to decide which activities are of relevance. An activity template is of the form:-

Active workspace I.D - Comparison
Status - Comparison

Thus an active workspace which was dependant upon *CS361_course* finishing would be of the form.

Active workspace I.D = *CS361_course*
Status = *Finished*

When an active workspace changes its status to match the activity template the activity is informed of this change.

8 Architectural Issues

The work described so far has centred on the development of a series of models giving different views of a CSCW environment. Between them, these models specify the functionality of the environment. The section considers how these models may be realised within a computational platform and defines a potential *architecture* for a distributed MOCCA environment. The architecture needs to meet two key requirements. First, it should provide some degree of *transparency* to applications and users. Second, it should provide inter-operability between applications running on different hardware and software platforms, at different locations and under different managements.

The goal of transparency is to hide dimensions of the system that are unnecessary for a particular application and so make applications easier to build and to manage. There are several different kinds of transparency to be considered. So far, MOCCA has identified a number of transparencies as being important for CSCW [NAVARRO 92b]. These transparencies and their meanings are summarized in figure 5. It is also important to note that, in some situations, applications and their users may need to be aware of what is going on within the environment. Given the flexibility of CSCW systems it is important that the concept of *selective transparency* is applied within cooperative environments [RODDEN 91] and that the degree of transparency can be selected and modified.

Transparency	Central Issue	Result of transparency
organization	The organizations to which objects are members	Objects do not have to deal with differences between organizations
time	The time dependence between cooperating users, i.e. synchronous vs asynchronous cooperation	System design independent of form of interaction
location	Local or remote localization of objects	Objects unaware of the location of other objects
domain	The interest on a small subset of objects, those pertaining to a domain	Objects are unaware of the existence of other unrelated objects
view	The application of different views per user. For example, WYSIWIS	Users are unaware of the different views of other users
activity	The coordination and complexity of cooperative activities	Objects are unaware of the mechanism for running activities involving the coordination of several objects
security	Security concerns	Objects are unaware of the application of the security policy

Figure 5: Possible Transparencies provided by the MOCCA Architecture

From the point of view of applications, the architecture should provide a set of well-defined interfaces through which different components can be accessed. From the point of view of the environment, applications should appear as opaque objects which are homogeneous in their use of Environment components. This is most likely to be achieved by basing the architecture on present and future international standards for distributed systems. For the present, this means the Open Systems Inter-connection (OSI) [ISO-OSI 84] suite of network standards which specify standard applications such as the X.500 Directory Service [ISO-DS 88] and X.400 Message Handling Service [CCITT-X.400 88]. In the future, it implies the use of the Open Distributed Processing (ODP) reference model (ODP-RM) and its various components [ISO-ODP 90]. Given the current fledgling state of ODP and the importance of influencing the standards as they emerge, the MOCCA work is particularly concerned with basing its architecture on the ODP-RM [NAVARRO 92a]. Previous research has addressed the use of standard OSI services such as the Directory to support CSCW [PRINZ 89].

The proposed architecture provides a set of loosely connected managers which provide appropriate portions of cooperative support. These managers are intended to abstract over the services provided by existing OSI and ODP platforms in a simple and versatile manner. The set of managers within MOCCA is intended to be extensible so that a MOCCA platform which realized this architecture could readily evolve as the nature of the cooperative work being supported altered. Two of the managers within the architecture are aimed at supporting the manipulation of storage or information. A distributed repository, an *information store*, for shared information places a significant role in the cooperative environment and realises aspects of the information model. This is augmented by the provision of an *organizational database* which allows organizational information to be shared and is a computational realisation of the organizational model. The general architecture is presented in figure 6. User access objects are used to represent users within the architecture.

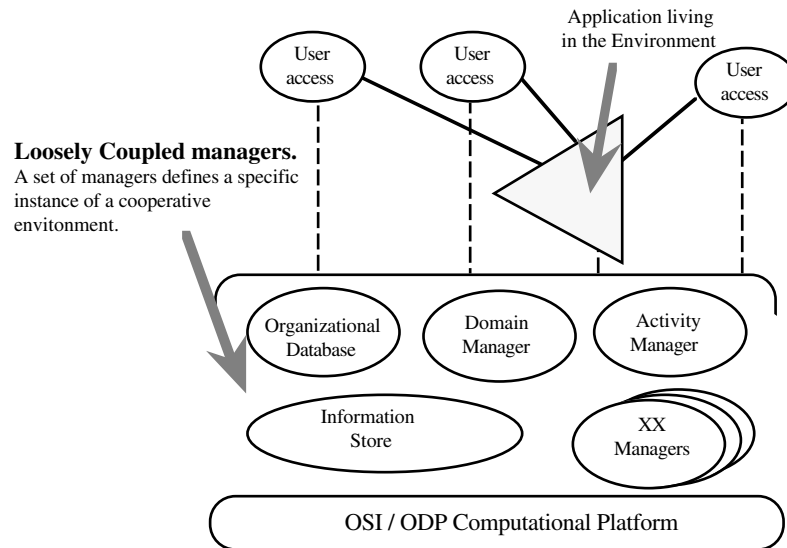


Figure 6: The MOCCA cooperative architecture

This architecture provides a set of optional interfaces for an application within the environment. This allows an application to maintain as many different ports as managers wants to interact with. The more ports are used, the more environment aware is the application. The extreme situation would be an application without any port to any manager: this application will have to contact to other objects. That service is provided by the domain manager, and supervised by the security manager. An architecture of this form allow services to be easily added and removed from the environment as future additional common services are identified. Not all MOCCA models canonically map onto managers within the environment. For example the workspaces are reflected in the support provided within the architecture by two components, *the domain manager* which provides the compositional functions for workspace and the *activity manager* which provides event propogation facilities and mainatins the status information for active workspaces.

The *Domain Manager* is an object to which all objects in the distributed environment have a connection (is a well known object/service). One of the services offered by the Domain Manager is the management of the compositional features of workspaces. The Domain manager also has the key importance for locating entities within the environment. It receives requests from objects requesting services (connection to other objects), it consults the Organizational Database and establishes connections between objects. It can also receives requests from other Domain Managers pertaining to different organizations to set-up inter-organizational workspaces. This inter-organizational operation may eventually provide support for bi-lateral agreements of electronic markets [MALONE 87b, ENGELBART 90]

The Activity Manager allows activities supported within an application to be registered outwith the application in the MOCCA Environment as active workspaces. Other applications interested in that active workspace can also register their interest in the status of that activity with the Activity Manager. The activity manger also realises the additional services required for event registration and propogation within the environment. The aim is to provide simple mechanism to promote the awareness of activities within the architecture. The detailed semantics of activities are not represented in the architecture but are an application issue.

9. Summary and future work

The goal of the MOCCA project has been to design an Environment which supports the management and development of CSCW applications within a common, framework. The first step was to identify a set of requirements for such an environment. These included support for information sharing between applications; provision of a common organizational context; facilities for navigating and locating objects; management of relationships between applications; encouraging users' awareness of collaborative activity through notification of events; and supporting background social interaction as well as more planned collaboration. In addition, the environment should be realizable within an open systems environment.

The main thrust of subsequent work has been in developing a set of models to represent different aspects of the environment. Collectively, these models specify the functionality of the environment. The rooms model provides a framework for the environment based on the notion of a virtual world of connected rooms through which people move and within which they interact with other people and various tools. The organization model maintains a description of the organizational structures and resources necessary to support CSCW applications and also the other models. The information model - gives mechanisms for representing the complex structures of communication information which is shared between CSCW applications. Workspaces provide a means for representing the externalised effects of group activities within the environment. The project has also outlined a distributed architecture for the implementation of these models as a set of system components which offer well-defined interfaces to applications. The architecture is loosely based on the Open Distributed Processing Reference Model for distributed systems in order to meet the requirement of establishing Open CSCW Systems.

At present, the requirements of the different MOCCA models are reasonably well defined and most of the groups effort is focused on specifying each model in detail. In addition, further models may be required to address areas of the CSCW Environment which have not yet been considered. Perhaps the most pressing of these is the development of a security model for CSCW applications - an area so far untouched. Beyond this, the detailed models could be used to specify the system components which are outlined in the distributed architecture and so implement the model on top of a suitable open platform (e.g. the ANSA platform for ODP [ANSA 89]). It would be a mistake to pretend that MOCCA has addressed all, or even most, of the issues involved in defining a CSCW environment. Instead, we think that our project represents a few steps in what is an important direction for CSCW.

References

- [ANSA 89] *The ANSA Reference Manual*, Architecture Projects Management Limited, Cambridge UK, 1989.
- [BANNON 91] L. Bannon and K Schmidt, *CSCW: Four Characters in Search of a Context*, in J.M Bowers and S.D. Benford (eds), *Studies in Computer Supported Cooperative Work: Theory Practice and Design*, North-Holland, Amsterdam, 1989.
- [BENEDIKT 91] Michael Benedikt, *Cyberspace - First Steps*, The MIT Press, 1981.
- [BENFORD 91] Steve Benford, *Building Group Communication on OSI*, Computer Networks and ISDN Systems, 23, PP 87-90, North-Holland.
- [BENFORD 92] Steve Benford, *A Standard for OSI Group Communication*, to be published in the International Journal of Computer Networks and ISDN Systems, North-Holland, 1992.
- [BIGNOLI 91] C. Bignoli and C. Simone, *AI Techniques for Supporting Human to Human Communication in CHAOS*, in *Studies in Computer Supported Cooperative Work: Theory, Practice and Design* (J. Bowers and S.Benford eds), North-Holland, 1991.

- [BORNING 91] A Borning and M Travers, *Two Approaches to Casual Interaction over Computer and Video Networks*, Proceedings of CHI '91, New Orleans, April 27-May 2 1991, pp13-19.
- [BOWERS 88] J. Bowers and J Churcher, *Local and Global Structuring of Computer Mediated Communication: Developing Linguistic Perspectives on CSCW in Cosmos*, Proceedings of CSCW88, Portland OR, 1988.
- [CCITT-X400 88] CCITT, *Message Handling Systems: System Model - Service Elements*, Recommendation X.400, 1988, (Also published as ISO International Standard 10021).
- [CLARKSON 91] Mark A. Clarkson, *An Easier Interface*, BYTE, February 1991, pp227-282.
- [DE CINDO 86] De Cindo F., De Michelis G., Simone C., Vassallo R. and Zanaboni A., *CHAOS as a Coordination Technology*, in Proceedings of CSCW86, Austin, Texas, 1986.
- [ENGELBART 90] D. Engelbart, *Knowledge-Domain Interoperability and an Open Hyperdocument System*, Proceedings of CSCW'90, Los Angeles, CA, ACM Press, 1990.
- [GALE 91] Stephen Gale, *Adding Audio and Video to an Office Environment*, in Studies in Computer Supported Cooperative Work: Theory Practice and Design, Bowers and Benford (eds), Elsevier Science Publishers, 1991.
- [GAVER 91] William W. Gaver et al., *Working Together in Media Space: CSCW Research at EuroPARC*, in Computer Supported Cooperative Work: The Multimedia and Networking Paradigm (Proceedings of the Omnicom Seminar, London, 16-17th July 1991), Omnicom Ltd.
- [HILTZ 85] Starr Roxanne Hiltz and Murray Turoff: *Structuring Computer-mediated Communication Systems to avoid Information Overload*. Communications of the ACM, July 1985, pp 680-689.
- [HORTON 87] Horton M.R., Adams R., *Standard for the Interchange of USENET Messages*, Network Information Center RFC 1036, SRI, California, 1987.
- [ISO-DS 88] ISO and CCITT, *Information Processing Systems - Open Systems Interconnection - The Directory*, ISO 9594-1-8, CCITT X.500-X.521, 1988.
- [ISO-GC 91] ISO/CCITT, *Group Communication Working Document X.gc (version 8)*, Output from the joint ISO/IEC/CCITT meeting at Ottawa, June 1991 (available from the authors).
- [ISO-ODP 90] ISO, *Information Processing Systems - Open Systems Interconnection (OSI) - Open Distributed Processing (ODP)*, ISO/TC 97/SC21/WG7, 1990.
- [ISO-OSI 84] International Organization for Standardisation, *Basic Reference Model for Open Systems Interconnection*, ISO 7498, 1984.
- [LOVSTRAND 91] Lennart Lovstrand, *Being Selectively Aware with the Khronika System*, Proceedings of ECSCW '91, Amsterdam Holland, Kluwer Academic Publishers, pp265-278.
- [MALONE 87a] Thomas W. Malone, Kenneth R. Grant, Kum-Yew Lai, Ramana Rao and David Rosenblitt, *Semi-structured Messages are Suprisingly Useful for Computer Supported Coordination*, ACM Transaction on Office Information Systems, Vol. 5, No. 2, pp 115-131, 1987.
- [MALONE 87b] T. Malone, R.I. Benjamin, J. Yates, *Electronic Markets and Electronic Hierarchies*, Communications of the ACM, Vol 30 No 6, 1987.
- [NAVARRO 92a] L. Navarro, W. Prinz, T. Rodden *Towards Open CSCW Systems*. to appear in: Proceedings of the IEEE Workshop on Future Trends of Distributed Computing Systems in the 1990's, Taipei, April '92.
- [NAVARRO 92b] L. Navarro, M. Medina, T. Rodden *Environment Support for Cooperative Working*. To appear in: Proceedings of the IFIP 6.5 Conference on Upper Layer Protocols and Applications, Vancouver, 1992.
- [PALME 90] Jacob Palme *SuperKOM - a distributed computer conference system*. Proceedings of the IFIP 6.5 Symposium on Message Handling Systems and Application Layer Communication Protocols, Zürich, October 1990, North-Holland.
- [PANKOKE 89] Uta Pankoke-Babatz (ed), *Computer Based Group Communication - The Amigo Activity Model.*, Ellis Horwood, 1989.
- [PANKOKE 89] Uta Pankoke-Babatz (ed), *Computer Based Group Communication - The Amigo Activity Model.*, Ellis Horwood, 1989.
- [PRINZ 89] Wolfgang Prinz, *Relevance of the X.500 Directory to CSCW Applications: Directory Support for Computer Based Group Communication*, in Proceedings of the First European Conference on Computer Supported Cooperative Work, Gatwick, London, 1989.
- [RODDEN 91] T. Rodden, G. Blair, *Distributed Systems and CSCW: the problem of control*, Proceedings of the IFIP International Conference on CSCW, Berlin 1991.
- [ROOT 88] R.W. Root, *Design of a Multi-Media Vehicle for Social Browsing*, Proceedings of CSCW '88, Portland, Oregon, Sept 26-28 1988, pp25-38.

- [SMITH 89] Hugh Smith, Julian Onions and Steve Benford, *Distributed Group Communication: the AMIGO Information Model*, Ellis Horwood, 1989.
- [SMITH 90] Hugh Smith, Steve Benford, Alan Shepherd and Howidy Howidy, *OSI Based Group Communication*, Proceedings of the IFIP 6.5 Conference on Message Handling Systems, Zurich, October 1990, North-Holland.
- [TREVOR 93] J. Trevor, T.Rodden, G. Blair, *A lightweight approach to activity support*.Submitted for ECSCW'93, Milano, September 1993.
- [WINOGRAD 86] T. Winnograd and F. Flores, *Understanding Computers and Cognition: A New Foundation for Design*, Norwood, New Jersey, Ablex 1986 and Addison-Wesley, 1987.