MODELLING GROUP COMMUNICATION STRUCTURES : Analysing four European projects

Pippa Hennessy (pah@cs.nott.ac.uk)
Steve Benford (sdb@cs.nott.ac.uk)

Department of Computer Science
University of Nottingham*

John Bowers (jmb@psyc.nott.ac.uk)

Department of Psychology
University of Nottingham

Abstract

This paper was first published in the Singapore International Conference on Networks '89, pages 56–61.
It describes the COSMOS, AMIGO MHS+, AMIGO Advanced, and MacAll II models of group communication, and compares them from four key perspectives. The need for an integrated approach to modelling group working is stated, and features that should be included in any model are identified.

1 INTRODUCTION

Recent years have witnessed a rapid growth in the use of computer based communication services such as electronic mail, bulletin boards and conferencing facilities. This trend is likely to accelerate with the introduction of new standardised services such as X.400. However, it is already recognised that existing services exhibit a number of critical limitations:

- Communication occurs on a one-to-one (i.e. person-to-person) basis or, occasionally, on a one-to-many basis (e.g. distribution lists in electronic mail). True group communication is not supported.
- Communication is unstructured. Little support is provided for supporting natural communication patterns and reducing the problem of *information overload* associated with current communication facilities.

These limitations have motivated a number of recent projects considering the issue of group communication or *Computer Supported Cooperative Work (CSCW)*. Support for CSCW facilities will be vital as the communications environment expands.

*University of Nottingham, University Park, Nottingham, NG7 2RD, England. Tel: +44 602 484848 ext 3702. Fax: +44 602 588138

The overall goal of this paper is to analyse some different models of group communication developed in four recent European research projects. The paper then suggests the need for an integrated approach combining the best features of each model. The four projects summarised are: COSMOS, AMIGO MHS+, AMIGO Advanced, and MacAll II. They have been chosen due to the scope of issues they address, the range of models developed, and the authors' close involvement, and therefore detailed knowledge, of each project.

This paper is structured as follows:

- Section 2 provides a quick overview of the major features of each project. In particular, it describes their abstract models of group communication.
- Section 3 compares these models from several perspectives.
- Section 4 suggests the need for an integrated approach combining and extending the best features of each model. It also lists some of the properties which this integrated model should have.

2 DESCRIPTION OF THE MODELS

2.1 THE COSMOS PROJECT

COSMOS is a multidisciplinary research project, funded by the Alvey initiative in information technology, involving participants in the UK from various backgrounds including universities and industry. The project was initiated to examine the general nature of problems associated with CSCW and, more specifically, to develop an abstract framework for the definition of group working practices which could be mapped to a wide variety of implementations, including those based on distributed computer systems. The central claim of COSMOS is that group working should be both *structured* and *configurable*.

There are five main areas of research within the project. First, the development of a *Structure Definition Language* (SDL) that allows a user to formally define a structure for group communication in a system independent manner. Second, *task analysis* of communication within various groups and the development of a *User's Conceptual Model* based on this work. Third, the specification and implementation of an X.400 based prototype COSMOS system supporting various aspects of SDL within a distributed computer system. Fourth, the design of user interfaces to a COSMOS system. Fifth, the development of methods for evaluating the usability of a COSMOS system. Only the first of these areas is relevant to this paper.

The COSMOS abstract model of group communication is embodied in the Structure Definition Language (SDL) [14]. In a COSMOS system, group working and the exchange of messages will be coordinated in terms of abstract *Communication Structures* (CSs). Each CS is defined in terms of SDL, and consists of a set of rules and other notations which specify how group communication is expected to unfold. A CS is an abstract representation that is independent of any particular occasion of use.

A CS contains the following components [2, 3]:

- Actions – These are divided into *exchanges*, which are elementary communicative acts involving at least two participant roles and usually one object; and *encapsulated actions* (EAs) which do not involve any exchange (e.g. creating objects).

- **Rules** – These define the conditions under which actions can occur.
- **Roles** – These are the agents who initiate actions. A role may be played by an individual user, a collective, or an automated process.
- **Objects** – These are pre-defined structured objects that are created and exchanged during the performance of activities.
- **Conditions** – These are expressions, resolving to *true* or *false*, which define the contexts in which rules are triggered.
- **Temporal order constraints** – These explicitly state constraints on the order in which events, including actions, should occur. The ordering of events may also be implicitly constrained by the structure of the rules.

A COSMOS Activity is instantiated by a user selecting the appropriate CS definition, and supplying the data needed to execute the definition (e.g. assigning users to roles). Activities progress through actions being performed by one or more defined roles, subject to constraints (rules and conditions) specified within the CS. Messages are exchanged with other users that are performing roles within the Activity, and users will be prompted when they are expected to perform an action. Depending on the form of the Activity, it will usually finish when all roles have completed their defined actions.

2.2 THE AMIGO MHS+ PROJECT

The AMIGO MHS+ project involved several European institutions, and was funded by the European Community under the Cost11-ter programme. The goal of the project was to consider the extension and integration of existing communication services such as *Electronic Mail* and *Directory Services* to support group communication [10, 11]. Another concern was to realise facilities offered by existing centralised group communication services (e.g. centralised bulletin board and conferencing systems) within a general distributed framework. As such, the project was not concerned with the development of models explicitly describing group communication patterns. Instead, it addressed the need for an abstract framework facilitating the realisation of such models within distributed systems.

The AMIGO MHS+ philosophy considers communication in terms of the shared access of information by groups. Rules for the explicit, ordered transfer of messages between the members of a group are not considered. Rather, group working is addressed in terms of the interaction between groups and their information within a number of *environments*.

The project developed a number of models describing information sharing at different levels of abstraction. First, an abstract datamodel describing the structure of, relationships between, and operations used to manipulate information. Second, an example architecture demonstrating how the datamodel might be supported by the coordination of distributed services such as electronic mail. Third, models for the extension of existing services (e.g. distribution lists within X.400) and the specification of new services (e.g. an Archive service) to support enhanced group communication features. Of these models, the datamodel is the most relevant to this paper and is examined in greater detail in the following paragraphs.

The AMIGO MHS+ datamodel [1] facilitates the abstract description of information shared within and between groups. Shared information is modelled in terms of three key elements:

- **Information Objects** - Each information object (IO) represents a logical named unit of information. There are two classes of IOs; *atomic objects* (such as messages and documents); and *compound objects*, which represent dynamic sets of information.
- **Operations** - Operations allow the manipulation of IOs. This includes their creation, retrieval and modification as well as sophisticated searching facilities.
- **Environments** - Environments define logical spaces within which IOs exist and operations are invoked. An Environment can be viewed as representing the area within which a group shares information.

The notion of compound objects is important. They contain a special *derivation rule* that, when *enumerated*, obtains the current members of the set (which may also be compound objects). These objects can represent complex and dynamic information structures in a natural manner. The concept of environments is also a major feature of the model. The environment plays two major roles. Firstly, it provides a shared work area associated with a group. Secondly, by identifying the resources associated with a group, environments bind the abstract datamodel to its realisation within a distributed system.

2.3 THE AMIGO ADVANCED PROJECT

The AMIGO Advanced project, as its name implies, was closely linked to the AMIGO MHS+ project. It also involved several European institutions and was funded by Cost11-ter. The project aimed to investigate the requirements of group communication tools, and ultimately to develop a model (the AMIGO Activity Model – AAM) based on the concept of *activities* representing group communication processes [9, 4]. The goal of the project was to develop an abstract model of communication which, unlike the AMIGO MHS+ model, is independent of current technology and services.

First, a description of group communication in general, and communication within an electronic environment in particular, was developed. This description was explicitly based on group communication procedures within the office environment. It was then used to build a computer-usable general specification means for describing group communication tools. The emphasis of the research was on the need to *support* the user in his/her use of any communication tool.

An activity within the AAM is defined as the *intention and goal of a group* (e.g. the joint editing of a paper); it has four basic components:

- Roles – Within each activity, a set of role classes is defined. Each member of the group must be associated with at least one role that is an instance of one of these classes. The class definitions include the functions that instances of this class of roles may perform or request to be performed, and the message object types that can be sent or received.

- **Functions** – These are the operations performed within group communication processes (e.g. send, reply, receive, forward). They are atomic in the sense that they must always be completed, and they must be executed entirely by one processor (a processor can be either a role instance or the system). The definition of a function includes its name, its input and output parameters, and the procedure to be executed.

- **Message objects** – These are the basic units of communication. Message object types are defined for each activity, specifying which attributes a message of that type will contain, and sometimes a range of values the attributes can take. Message objects may be associated with a *meta-message* containing a description of the semantics of parts of the message, the history of the message, and an overall context description.

- **Rules** – These are used to define how each role is expected to behave during the execution of an activity. If the activity requires that different phases are set up, this is achieved by changing the rule set. It is also possible to define sub-tasks of an activity using sets of rules.

An *activity template* definition is created by describing the above components. In addition to these, the external relations of an activity can be specified. This involves specifying the influence other communication processes have on the group activity (and vice versa).

The performance of an activity involves the creation of an instance from the appropriate activity template. In addition, instances of the roles and message objects are created. The performance of the activity is controlled by a *central coordinating entity* that uses the activity template to ensure that functions are executed as required.

2.4 THE MacAll II PROJECT

This project involved members of the Communications Research Group at Nottingham University, and was funded by the Digital Equipment Corporation. Its aim was to model communication processes within an office environment, and to implement a prototype support system based on this model.

The MacAll II project [12] was primarily concerned with group communication based on office procedures. Due to the bureaucratic nature of organisations, the types of activities considered typically involved filling forms and similar procedures. Consequently, a procedural approach was adopted, and messages were considered to be highly active objects that were constantly changing. It was decided to base the model (the Activity Model Environment – AME) on the activity model developed by the AMIGO Advanced Group, extending this to incorporate ideas that had evolved from a previous project (MacAll I [13]), for example, the concept of workspaces.

As in the AMIGO Advanced AAM, the central concept is that of an activity. Activities are the basic processes that happen in offices and each one is initiated and performed in order to achieve a specific goal. Unlike AAM activites, rules governing the performance of AME activities are implicit within the definitions of other components. The components of the AME are:

- **People** – These are only represented to the extent of the roles they can play.

- **Roles** – A representation of what people do within the office when they are involved in a specific activity.
- **Messages** – These are objects (e.g. memos, forms) used to transfer information between role instances They are *active* in the sense that they contain rules which dictate how they should be processed.
- **Iunits** – *information units* – These can be logical parts of messages (i.e. sets of fields that are usually filled in by one role), documents, etc.
- **Workspaces** – Conceptual work areas associated with roles where messages are received, work is done and items are stored.
- **Rules** – These define what is expected of a person playing a role, and the actions workspaces and messages take.
- **Rulesets** – Groups of names of rules that are always invoked together.
- **Activities** – The main function of these is to point to groups of objects that are all involved in the same activity.
- **Subactivities** – Logical groupings of tasks within activities.
- **The Organisational Manual** – This contains definitions of the above components.

During the performance of an activity, people are assigned to roles, and they follow the rules associated with those roles. Messages are created, and their rules direct their transfer between workspaces. When a message arrives at a workspace, the workspace notifies the appropriate role instances that a new message has arrived, and one of the role instances processes the message according to its rules.

3 DISCUSSION OF THE MODELS

This section presents a brief analysis of the four models described above. Its purpose is to draw out their important features and ascertain their major differences, thus providing input for a more complete, integrated model of group communication. The four models differ somewhat in goals and scope, and between them they address a range of issues from the abstract to the concrete. However, they all include the provision of a general framework for describing group communication. Consequently, having noted their different scopes, comparison of the models is a valid exercise. The models are compared from four key perspectives:

- Comparison of their general approaches to group communication.
- Comparison of their treatment of objects representing information.
- Analysis of their support for information domains.
- Comparison of the "activity" concept within each model.

3.1 GENERAL APPROACHES TO GROUP COMMUNICATION

3.1.1 Procedural vs. information sharing

This paper identifies two general approaches to the modelling of group communication, evident in the models considered. These may be loosely termed the *procedural* and *information sharing* approaches.

- The procedural approach considers the ordered flow of information between roles as the basis for communication within groups.
- The information sharing approach places the emphasis on a group working towards a common goal using shared information that is available to those who require it.

The difference between these approaches lies in the way group working is viewed. When using a procedural approach, information is seen as moving in some way between members of a group. Information objects (which include messages) are always changing and new data are included as an activity progresses. The model resulting from such an approach tends to be an outline of sequences of actions, and a description of the information objects that are acted upon and the agents that act on them. In particular, procedural models emphasise the temporal (and sometimes causal) relationships between communication acts.

The information sharing approach focuses more on the access of shared information objects by group members. These objects may have complex structures and may be grouped into sets reflecting the dynamic aspects of information. Group members access information objects via operations which support their retrieval and management. In addition, complex searching and structuring facilities are usually available. A model resulting from this approach has a highly developed concept of an information object, and communication is implicitly described in terms of which objects may be accessed by whom as opposed to explicit exchanges. The temporal relationships between events are not always explicitly represented in these models.

3.1.2 The approaches adopted by each project

In reality, the models considered by this paper form a spectrum with the procedural and information sharing approaches at either end.

- The AMIGO Advanced model is strongly oriented towards the procedural approach with a rule based model centered on the explicit exchange of messages by roles, within activities.
- COSMOS is based on the procedural approach, indicated by the use of production-like rules within SDL. However, some support for information sharing is provided by information servers in the COSMOS prototype.
- The MacAll model combines both approaches, adopting a fundamentally procedural model with limited support for information sharing within *workspaces*.
- The AMIGO MHS+ model adopts the information sharing approach with the datamodel as its cornerstone.

The examples of group communication explored by each project reflect the different approaches they adopt. COSMOS and AMIGO Advanced examine structured, well ordered activities such as "voting". The MacAll project primarily concerns office communication, considering form-based activities such as "trip planning". AMIGO MHS+ considers less structured activities such as "conferencing" where the ordering of events is not critical.

3.1.3 Shortcomings of each approach

Both the procedural and information sharing approaches exhibit crucial limitations and this paper proposes that neither can adequately describe all common classes of group communication. The main limitation of the procedural approach is its lack of support for group communication activities in which the structure of shared information is important. In particular:

- It is difficult to identify and construct sets of information from those objects exchanged between roles (e.g. accessing an entire conversation as a sequence of exchanged messages).
- Little provision is made for exploiting the relationships between information (e.g. "give me all messages commenting on document X" or "all messages sent by Y or Z").
- Methods for accessing the history of past activities are primitive

On the other hand, the lack of temporal structure within the information sharing approach renders it unsuitable for describing classes of communication in which the order of events is critical. In particular:

- There are no temporal constraints on actions.
- There is no concept of *role* supporting both users and the system in achieving specific goals, or reflecting the division of labour that is common to may group working activities.
- Rules for the processing of information cannot be specified (e.g. "count the number of votes").
- The lack of an *activity* concept means that *patterns* of communication are not explicitly represented within the system.

3.2 THE TREATMENT OF OBJECTS

Each of the four models reviewed in this paper considers communication in terms of access to, or transfer of, *objects* containing information. This section compares the definition and treatment of objects within each model. In particular, the models can be contrasted in terms of their support for *structured objects*, *active objects*, and *sets of objects.*

COSMOS objects COSMOS objects are the items transferred between roles in an *exchange* and created as the result of *encapsulated actions.* They may be ephemeral, such as messages which are exchanged once, or more durable, such as documents exchanged many times.

The structure and contents of COSMOS objects are not considered of importance to the definition of SDL communication structures and are therefore not defined in detail. However, two methods are provided for referencing objects within SDL: by subscript (e.g. paper(3)), or by history (e.g. the paper which role A sent to role B). COSMOS objects are not active (i.e. do not contain rules for their own processing). Instead, objects are processed according to the rule components of SDL communication structures. SDL allows the representation of relations between objects, where, for example, one object (e.g. an annual report) is based on or composed of others (e.g. departmental reports). The subscript and history mechanisms described above can also be used to exploit relations between objects.

AMIGO MHS+ objects The AMIGO MHS+ datamodel supports a highly developed object concept. Information Objects form the basis of the datamodel, and their structure and manipulation is the key to the AMIGO MHS+ view of group working.

AMIGO MHS+ Information Objects consist of sets of attributes where each attribute has a type (e.g. "author") and an arbitrarily complex value (e.g. "Joanna Bloggs"). The values of attributes may themselves be sets of attributes, thus allowing nested objects. This functionality is considered necessary for representing complex items such as encapsulated messages. Each Information Object is identified by a globally unique *distinguished name* consisting of a subset of its attributes. Information Objects are not active. The datamodel allows objects to be grouped into sets via the compound object construct. These sets may be highly dynamic (e.g. "all messages sent by Joanna Bloggs") and operations are provided to access members of the set at any one time. Furthermore, transient sets may be identified within search operations. The ability to construct general dynamic sets of Information Objects is a major feature of the AMIGO MHS+ datamodel.

AMIGO Advanced objects In the Activity Model, message objects are defined as those objects that are exchanged between communication partners within activities, whose primary purpose is to communicate information. They may contain instructions to the receiver that specify what s/he is expected to do as a result of receiving the message object. Another optional component is the *meta-message*, this contains history and context information.

The structure of a message object is defined as a matrix. The first column contains a list of attributes the message has, the second contains the default or initial values for these, and the third column contains names of functions that can affect the attributes. Message objects are not active. Relationships between message objects, or parts of message objects, can be explicitly specified (e.g. reply, response, version of, copy of). Relations can also be specified that define logical groupings of message objects.

MacAll objects MacAll objects are of two types; messages and iunits. Messages are used for transferring information between, and collecting information from role instances. Iunits, or *information units*, are, as their name suggests, atomic units of information. They may exist as part of a message, or they may be distinct objects stored within workspaces. Both types of objects have a *history* associated with them.

Messages consist of a set of iunits, which in turn comprise a group of fields. An iunit within a message is defined as the group of fields that is expected to be filled by one role. Messages are *semi-structured*, in that fields are typed, but may be given the type "text", which means they may contain anything. Iunits are structured if they are parts of messages, and may be structured otherwise. Both messages and iunits are active objects, and contain rules for their own processing. There are no explicit mechanisms for grouping objects into sets, but rules may be defined that allow the refining of a group of messages within a workspace according to criteria based on their history or values of fields. In addition, a field may contain another iunit or message, so objects can be grouped in this way.

3.3 SUPPORT FOR INFORMATION DOMAINS

This section examines the support within each model for the concept of *information domains*. Information domains describe spaces in which the information pertinent to a group is located and managed. Furthermore, they contain the *resources* (e.g. agents and services) required by users playing various roles within a group. There are a number of key questions to be considered when modelling information domains:

- How should the boundaries of information domains (and groups) be defined?
- What properties should these boundaries have. For example, how flexible are they, and can they overlap?
- What effect do information domains have on other components of the model. For example, are objects given local names for each environment they can exist in, or do they have global names?

Both the AMIGO MHS+ and the MacAll projects support information domains to some extent, and address some of the above questions. This support is considered in the following sections. The domain concept is less evident in the COSMOS and AMIGO Advanced models.

MacAll "workspaces" In the MacAll model, people interact with the system by assuming *roles* and thus becoming *instances* of roles. A role consists of a pre-defined set of rules that any person assuming that role is expected to follow. Each role has a *workspace* associated with it that may be accessed by all instances of that role, and instances of other roles may have some limited or unlimited access to the contents of the workspace. A workspace contains:

- **Description** – A textual description of the workspace.
- **List of role instances** – Every time an instance of a role is created, it registers with the appropriate workspace; thus each workspace has a list of all current instances of its associated role.
- **Trays** – An *in tray* which contains incoming messages, an *in progress tray* that contains messages that are being processed, serving as a locking device, and an *out tray* which

is where messages that have been processed by a role instance wait while they perform any self-processing that is required by their rules.

- **Resources** – These are tools that a role instance may need to perform his/her role. This includes basic tools such as calculators and calendars, and also means of accessing external devices such as printers.
- **Filing Cabinet** – This contains archived messages and iunits that may be needed by role instances.

Conceptually, a person does all the work associated with the roles that they play in the workspaces appropriate to those roles. Within these workspaces, all resources needed to perform any task that may be required of the associated role will be available. Workspaces are strictly defined as conceptual work areas associated with one specific role; boundaries may be transparent but do not overlap.

AMIGO MHS+ "environments" The AMIGO MHS+ project supports information domains in terms of *environments*. These serve two purposes. First, they describe a locus of effect for operations such as searching. Constraining the effects of such operations will be critical where truly world-wide communication occurs. Second, they indicate the resources associated with a group and therefore supporting the realisation of conceptual operations within a distributed system. For example, a search operation can be targeted at specific agents in a storage system as identified within the group environment.

The description of an environment may include:

- A textual description of the environment.
- A description of the groups of communication entities within the environment.
- Access control rules that govern access to information associated with the environment.
- The names of resources that may be used within the environment.
- The names of important information objects within the environment.
- A description of the organisation of information objects within the environment.

Environments as defined by the AMIGO MHS+ project are more general and flexible than the MacAll workspaces. In particular, they may overlap and objects may belong to more than one environment. Overlapping environments are particularly important because they support the sharing of information between, as well as within, groups.

3.4 ACTIVITIES

It is generally accepted that the concept of an *activity* is important when modelling group communication processes. It can be loosely described as the events that take place when a group of people is formed in order to achieve a common goal. There are a number of issues involved in the modelling of activities:

- Role-oriented vs. mediator-oriented approach.
- Relationships between activities.
- Error-handling and re-trying mechanisms.

This section clarifies the distinction between role-oriented and mediator-oriented activities, and then examines the way each model has dealt with the modelling of activities.

3.4.1 Approaches to the modelling of activities

It is possible to approach the modelling of activities from two perspectives; *role-oriented*, and *mediator-oriented*. This distinction was highlighted by researchers in the AMIGO Advanced project.

A role-oriented approach looks at the activity from the viewpoint of the entities involved in its performance; and rules that govern events are specified within the definitions of these entities. The main advantage of this approach is that the distributed nature of specifications clarifies the division of tasks between roles, and enables local support for each role-player to be provided. However, due to a lack of co-ordination, tracking objects and dealing with unexpected behaviour is difficult.

On the other hand, the mediator-oriented approach views the activity as a distinct entity in its own right. Activities and the rules for their performance are separately defined. The advantage of this approach is that some agent somewhere will be co-ordinating events, which facilitates exception-handling. It also makes tracking objects and checking on the progress of the activity much easier. The disadvantage is that there is an inherent lack of support for role-players.

3.4.2 The treatment of activities by the models

COSMOS activities: The COSMOS model takes the mediator-oriented approach towards modelling activities. A *communication structure* (CS) is defined, which is a structure of production rules that specifies the order that events should take place within an activity. Actions associated with entities (roles and objects) are implicitly specified within these definitions, rather than being explicitly listed in association with the appropriate entity.

Relations between activities can be explicitly specified. Conditions that must be met before actions are performed can involve roles or objects from other CSs; objects from other CSs can be involved in actions, and an action can explicitly require a role to instantiate another CS.

At present, no attempt is made to specify or allow the specification of error-handling or re-trying mechanisms.

AMIGO MHS+ activities: No definition of activities as such was developed by this project; because of the importance attached to the control of information objects and storage and access to these, little attention was paid to the specification of the way this information should be used to achieve group goals.

AMIGO Advanced activities: In the AMIGO Advanced activity model, activities are modelled from a mediator-oriented point of view. Unlike COSMOS activities, each component is specified separately within an activity definition, and the rule component defines the way these interact. When an activity is instantiated, a central coordinating entity (CCE) directs performance of the component tasks.

Relations may be established between activities either by the invocation of one activity by another, or by explicit co-operation between activities. In the first of these, the invoked activity is treated as a role-player in the *parent* activity; whereas in the second, there is communication between role-players in each of two activities.

As a default, when an error occurs, a human administrator is informed and asked to resolve the problem. However, an activity definition may include information that allows the CCE to resolve situations where expected events have not occurred. If an unknown event occurs, a human is always informed.

MacAll activities: AME activities were developed using a primarily role-oriented approach. All rules governing the performance of activities are contained within objects (roles, messages, iunits, workspaces). However, activity objects are specified. The purpose of these is to act as a means of tracking particular object instances involved in activity instances. They also contain a textual description of the activity which may be used as a source of reference by role-players who wish to obtain an overall view of the activity.

Relations between activities are not specified.

The notion of *subactivities* was introduced as a primitive means of error-handling. These are groupings of tasks within an activity, and if any part of the activity needs to be re-done, it can theoretically be performed again in isolation.

4 CONCLUSION: an integrated approach

The previous sections have described and analysed four models of group communication. Although a wide range of issues has been addressed by these models, no single model has dealt with them all. This is, in part, due to differences between their general approaches towards modelling group communication.

This paper proposes that an integrated approach to modelling is needed in order to develop a comprehensive model of group communication structures. Furthermore, "cobbling together" the best features of current models is not the optimal means of developing this model. Instead, the new model should be based on careful consideration of all issues involved in the modelling of group communication. This may result in the use of previously developed solutions. It may also involve the examination and reconceptualisation of issues at a more basic level.

The full description of this approach to modelling, and a specification of requirements for a resulting model, are clearly for further study. However, based on the preceding review of four current approaches and models, we feel that we can identify some features that should be exhibited by such an integrated model:

- The model should be developed using a primarily procedural approach.

- A general object model should be used to represent both information objects and other components of the model.

- A well-developed concept of *information domains* should be included.

- The concept of *activities* should be well-defined, and relations between activities should be specifiable and configurable.

A primarily procedural approach should be used when developing a model of group communication. The main reason for this is that communication processes are naturally described in terms of procedures, that is, sequences of events. In other words, procedural models effectively capture the action-oriented nature of communication. Also, the shortcomings of a purely information sharing approach (lack of support for temporal constraints and rules for information processing, lack of *role* and *activity* concepts) do not arise when using a procedural approach. However, some aspects of information handling need to be considered, and therefore the use of a general object model and the concept of information domains should also be features of the model.

The use of an object model (e.g. object-oriented programming languages such as Smalltalk [6], the CCITT X.407 standard [8]) as a basis for specifying information objects, and other components of the model, gives an underlying structure for these entities, and allows relations between them to be specified. Objects could be grouped into sets by using inheritance mechanisms, and compound objects could be supported. Clearly, an integrated model could draw on other developments in object-oriented programming languages (e.g. active objects could be used to represent *intelligent messages* [7]).

The concept of information domains is an important feature of an integrated model for three reasons. First, it is the means by which an abstract model can be associated with actual services and resources available to users of the system. Secondly, an information domain is useful for grouping related information objects, and controlling access to, and defining the locus of, operations on these objects. Finally, roles and people can be grouped within these domains, and relations between groups can be represented by overlapping and nested domains.

Activities are used to represent the goals and tasks of group working. The concept of an activity in some form is the basis for most procedural models. However, the concept has yet to be fully explored and defined. Issues for further study include exception-handling, relations between activities, and partial definition of activities. In addition, a clear concept of an activity that includes the best points of role-oriented and mediator-oriented approaches is needed. Exactly how activities should relate to group goals and plans also needs investigation. It is possible that recent work in Artificial Intelligence on planning in multi-agent systems (e.g. [5]) will be a useful source of inspiration here.

To conclude, this paper has described the research of four European projects (COSMOS, AMIGO MHS+, AMIGO Advanced, and MacAll II) that have investigated the modelling of group communication. The resulting models have been compared from four key perspectives, and this has led to the conclusion that an integrated approach is required. Some of the features that such an approach should exhibit have been indicated, and areas for further study have been highlighted.

References

[1] S.D. Benford. A data model for the AMIGO communications environment. In R. Speth, editor, *EUTECO '88 - Research into networks and distributed applications*, pages 97–110, 1988.

[2] J. Bowers and J. Churcher. Local and global structuring of computer mediated communication: Developing linguistic perspectives on CSCW in Cosmos. In *Proc. CSCW '88*, pages 125–139, 1988.

[3] J. Bowers, J. Churcher, and T. Roberts. Structuring computer-mediated communication in COSMOS. In R. Speth (ed), *EUTECO '88 - Research into networks and distributed applications*, pages 195–210, 1988.

[4] T. Danielsen and U. Pankoke-Babatz. The AMIGO activity model. In R. Speth (ed), *EUTECO '88 - Research into networks and distributed applications*, pages 227–242, 1988.

[5] J. Doran. Action co-ordination in multi-agent planning systems. In G.N. Gilbert and C. Heath (eds), *Artificial intelligence and social theory*. MacMillan, 1983.

[6] A. Goldberg and D. Robson. *SMALLTALK-80: The language and its implementation.* Addison-Wesley, 1983.

[7] J. Hogg. Intelligent message systems. In D. Tsichritzis (ed), *Office automation*, pages 113–134. Springer-Verlag, 1985.

[8] *ISO/IEC draft international standard 10021-3*, 1988. Joint ISO/CCITT paper, version 6.

[9] U. Pankoke-Babatz (ed), T. Danielsen, P.A. Pays, W. Prinz, and R. Speth. *Computer based group communication - The AMIGO activity model.* Ellis Horwood, 1989.

[10] H. Smith, J. Onions, and S. Benford (eds). *Distributed group communication - the AMIGO information model.* Ellis Horwood, 1989.

[11] H.T. Smith. The requirements for group communication services. In R. Speth (ed), *EUTECO '88 - Research into networks and distributed applications*, pages 89–96, 1988.

[12] H.T. Smith, P.A. Hennessy, and G.A. Lunt. The Activity Model Environment: an object-oriented framework for describing organisational communication. In *Proceedings of the 1st European Conference on CSCW '89*, 1989.

[13] H.T. Smith, G.A. Lunt, D. Young, and J. Lawrence. MacAll: An interface design project. Final report on the MacAll I project, available from CRG, Computer Science Department, Nottingham University, 1987.

[14] R.E. Young (ed). *Interim report on the Cosmos project*, 1988. Issued by: Cosmos Coordinator's Office, Queen Mary College, London.