REQUIREMENTS OF ACTIVITY MANAGEMENT

Steve Benford

Communications Research Group
Department of Computer Science
University of Nottingham
Nottingham NG7 2RD, England

ABSTRACT

Recent years have witnessed the development of several theoretical models of cooperative group working based on the concept of role playing within structured *activities*. This paper proposes that the widespread implementation and adoption of products based on these models is critically dependent on a framework for managing and integrating activities within working environments. The term *activity management* is introduced to refer to the process of planning, administrating and executing activities according to the various management policies defined within local environments. The requirements of activity management are then discussed with particular emphasis on two key concepts: the use of a high-level notation for expressing management requirements and the need for an activity management architecture supporting the management of activities within distributed computer systems. The overall goal of this paper is therefore to define a program of research to progress activity based models of group working towards viable and useful implementations.

1. Introduction

Research into Computer Supported Cooperative Work involves the study of a wide range of disciplines. At one end of the scale, research into human factors and theories of human communication have led to the development of theoretical models for describing structures or patterns of group communication [1]. These models aim to provide a mechanism, often a language, for describing different patterns of group working. At the other end, the enhancement and extension of existing communication technologies is leading towards the development of new CSCW services providing far richer support for cooperative working than is currently available [2].

A common motivation behind much of this research appears to be the development of generic tools enabling the design and configuration of a wide variety of communication systems. For example, in defining their *Structure Definition Language* (SDL), the COSMOS project has aimed for a mechanism whereby users can configure their own communication [3]. The AMIGO Advanced project also purposefully describe the application of their *Amigo Activity Model* (AAM) in "designing concrete group communication activities" [4]. However, in order to realise these ambitions, several issues have to be addressed concerning the relationship between activities* and the environments within which they occur:

*Although this paper recognises that Amigo and Cosmos activities are not identical concepts, the conclusions of this paper are applicable to both models. Thus, the term is used to refer to both Amigo and Cosmos activities. For a comparison of the two see [1].

- Activities do not exist in isolation. Instead, they may be related in a variety of ways. For example, multiple activities might share the same resources or might be assigned relative priorities.
- Mechanisms for dynamic role and resource allocation need to be defined so that optimal use is made of an organisation's human and system resources when shared between a set of on-going activities.
- In order to design or configure new activities, information describing the behaviour of current activities must be gathered and analysed. For example, productivity and performance indicators or identification of bottlenecks and error analysis.

These are just a few of the issues to be addressed when integrating activity based systems within office, and other, environments. More generally, they fall under the banner of *activity management*, a term referring to the support required for planning, administrating and controlling activities within a variety of working environments.

The general requirement for activity management has been noted within recent literature and several groups have provided an initial outline of the required functionality [4, 5, 6]. However, up to now the development of activity based models has mainly concentrated on the internal structure of activities (i.e. descriptions of roles, rules and messages) as opposed to the relationships between activities and the external world. Without extensive support for activity management, activity based systems are unlikely to be widely accepted within the commercial world where tasks and projects typically occur within a structured environment. For example, issues such as task priorities, scheduling and job allocation are often subject to well defined management policies. Activity management addresses another important commercial requirement, namely the need for status and performance information enabling strategic management decisions to be made. Statistics are required for the control of current activities, the planning of future activities and the development of new group communication applications.

The primary goal of this paper is therefore to define the requirements of activity management with the intention of bridging the gap between the developing activity based models of group working and real world technologies and working environments. In doing so, the paper develops two key concepts:

- The use of a high-level notation for expressing management requirements.
- The need for an activity management architecture supporting the management of activities within distributed computer systems.

The remainder of this paper is structured in the following way:

- Section two defines activity management in terms of its objectives, timescales and management functional areas.
- Section three describes the requirement for a high-level notation for expressing activity management policies and defines some of the functionality which such a notation should provide.
- Section four outlines a distributed architecture for activity management enabling this functionality to be realised within distributed computer systems.

2. Objectives of activity management

The term *activity management* was introduced in the previous section to describe the relationships between activities and the environments within which they occur. This section presents a more detailed definition of activity management by first examining its objectives and timescales and second identifying a range of management "functional areas".

The underlying model assumed by this paper is that group working can be structured in terms of a range of activities occurring within various working "environments" which represent the humans, resources, projects and groups within organisations (the term environment is used informally here, but is defined more rigorously in section four). Activity management describes the control and monitoring of these activities according to a set of environment wide management policies.

2.1. Short, medium and long term goals

By refining the objectives of more general network and distributed systems management [7], a range of objectives for activity management can been identified. These may be conveniently characterised according to whether they define short, medium or long term goals.

- **Operational control** concerns the immediate run-time control of activities in order to maintain adequate levels of service. Critical issues include error handling, reporting and recovery as well as immediate performance monitoring and control.
- **Administration** describes a range of medium term issues concerning the configuration of activities and their binding to humans and resources. For example, role allocation, naming, prioritising and scheduling.
- **Analysis and planning** concerns the gathering and analysis of information describing the long term behaviour of activities. This supports the planning of new management policies, strategies for acquiring new resources and the design of new activities. The latter is particularly important in the context of "configurable" systems where there is a requirement for feedback into the design process.

The broad objectives and timescales of activity management are summarised by the following table.

Activity management timescales		
objective	timescale	examples
operational control	seconds-minutes	run-time error handling performance monitoring
administration	minutes-hours	role and resource allocation prioritising and scheduling
planning and analysis	days-months	performance evaluation fault analysis, accounting

A study of existing activity based systems reveals at least partial support for the administration of activities (e.g. procedures for instantiation) and also some mention of operational control [4, 8]. However, these systems appear to provide little support for the analysis and planning of activities.

2.2. Management functional areas

In addition to these general objectives and timescales, activity management may be defined in terms of several *management functional areas*. Management functional areas have been introduced into models of network management as a method of partitioning the huge range of possible management functions into a more manageable set of topics [9]. This technique would appear to be equally applicable to activity management. The following paragraphs examine the relevance of network management functional areas to a the specific management of activities.

Management issues may be divided into five functional areas.

Management Functional Areas
Configuration and naming
Performance
Security
Fault and error
Accounting

Configuration and naming management is primarily concerned with binding the components of an activity to the humans and resources present within an environment. This is more commonly known as "instantiation" and involves a wide range of issues such as naming the activity, allocating roles and system agents. The key aspect of configuration management is dynamically controlling the sharing of resources between multiple activities. Thus, configuration management goes beyond instantiation to consider the reconfiguration of an activity during its lifetime.

Performance management involves monitoring and maintaining the performance of activities. Network performance issues include speed, throughput and utilisation. However, there is no obvious corresponding general measure of performance for activities as a whole and different types of activity are likely to be subject to different performance criteria. Furthermore, a distinction has to be made between the performance of the underlying system, the subject of systems and network management, and activity specific measures of performance involving humans. Activity performance might be better measured in terms of *productivity*, a term which implies greater consideration for the human aspects of group working. Research into productivity and computer systems might provide valuable input to this topic [10].

Security management covers a range of security related issues. For example, policies for the authentication of roles (e.g. are public key techniques required or would encrypted passwords suffice?), specifying encryption techniques for messages and managing access controls which govern visibility of activities, roles and information (e.g. providing support for private activities). Security within communications systems is an established research topic and a range of literature exists which might provide useful input to research into activity specific management [11, 12].

Fault and error management involves specifying policies for identifying classes of error, specifying error handling mechanisms and describing error logging and reporting techniques. A key issue to be resolved concerns the relationship between errors and *defeasibility*. One of the design goals of the Cosmos Structure Definition Language was that rules within activities should be defeasible, that is overturnable in certain contexts without prejudice [3]. In other words, a deliberate choice to break the rules of an activity should not be regarded as an error in the sense of the failure of a piece of software of hardware. The same distinction might also be applied to "unconsumed actions" within activities. Research into activity fault and error management should address this distinction, perhaps via the further development of existing mechanisms supporting defeasibility (e.g. SPACE in SDL).

Accounting management is concerned with management policies for accounting and billing. This includes specifying who is responsible for payment and which algorithms are to be used for charging. It is likely that accounting policies will vary greatly with both activity and environment. Accounting management is likely to be a critical issue within commercial systems.

In summary, this section has refined the concept of activity management by considering management in terms of a range of timescales and functional areas. The following section proposes the development of a notation for expressing activity management policies and presents examples of how such a notation might support these issues.

3. A notation for activity management

3.1. Initial inputs to an activity

A study of current activity models and implementations reveals that the following general steps are required in order to initiate an activity (see figure one).

- Specification of the activity's *communication structure* (Cosmos) or *activity template* (Amigo Advanced) using a formal notation such as SDL or the (AAM). This defines the *rules, roles, functions/actions* and *message objects* which combine to implement the activity.
- Instantiation of the activity using knowledge of local organisational structure and systems. For example: descriptions of people, their roles and responsibilities; descriptions of equipment and systems; descriptions of departments, projects and groups.

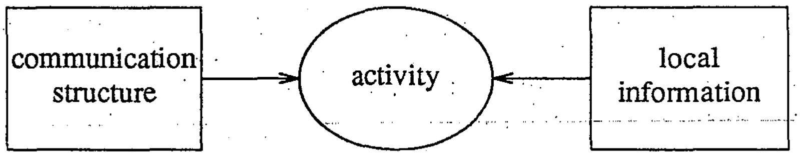

Figure 1: Inputs to an activity

This paper proposes that, in order to meet the objectives of activity management, a third input is required in the form of local activity management policies (see figure two). These provide a high-level specification of how the elements of the communication structure should be mapped onto the humans and resources in the local environment and also how the activity should be monitored and controlled. As an example of a "high-level" policy, the binding of a role might be described in terms of the general characteristics that the role-player should possess. Local activity management policies should facilitate the full range of management functionality specified in the previous section.

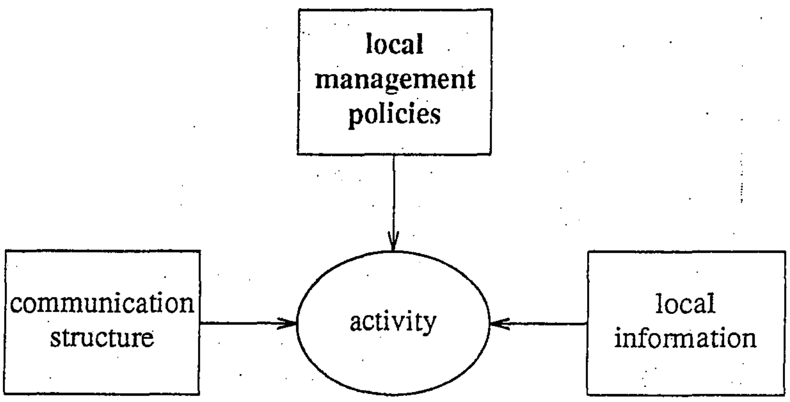

Figure 2: Inclusion of management policies within an activity

It is important to understand the distinction between these different inputs.

- The **communication structure** provides a **global** template for the activity which is independent of the local environment. For example, the general structure of a "conference" can be defined across a range of environments.
- The **local management policies** provide a high-level specification of how the activity is to be run within a specific environment and determine its relation to other activities within that environment. Thus, management policies specify the **local** aspects of an activity. For example, conference security and accounting might be subject to different requirements within different environments.
- Local information describing the structure and resources available within an organisation is contained within the local *organisational information base.*

The distinction between global structures/patterns and local management policies is one of the key points made by this paper.

3.2. What is meant by a "management policy"

Before progressing further, it is necessary to clarify the meaning of the term "management policy" as used in this paper. A management policy is a statement of how some management function is to be performed. For example, a policy might specify how a role is to be configured by providing a list of names for role-players or, more generally, a description of their characteristics. The policy should be abstract enough to shield the policy maker from minutiae such as managing specific network addresses, and yet be concrete enough so that it can be automatically interpreted by an automated management system.

Strategic organisational policies which might be defined by management (e.g. reduce production costs by 10%, weekly reports to be published by 9:00 AM each Monday) are not covered by the term "management policy". Implementing such strategic policies requires human decision making. Instead, the goal of this paper is to provide management with a tool which i) supports strategic decision making by providing relevant statistical information and ii) implements specific policies arising from strategic decisions. Perhaps the management policies outlined in this paper would better be described as "medium-level" in the sense that they bridge the gap between very high-level strategic policies and very specific low-level functions.

In summary, humans make and interpret strategic and political decisions. The activity management system interprets specific management policies arising from these decisions.

3.3. A formal notation

Several formal notations exist for specifying communication structures/patterns within activities (e.g. SDL, AAM). These define the objects to be managed for each activity in terms such as *rules, roles, functions/actions* and *message objects*. This paper proposes that a formal notation for expressing activity management policies is also required and, furthermore, that such a notation could be developed as an extension or annotation of these existing notations.

The specification of a formal notation for activity management is a topic requiring extensive research and is likely to require a detailed analysis, and perhaps further development, of existing notations for specifying activities themselves. Bearing this in mind, the following paragraphs provide only a brief sketch of how such a notation might be developed. The reader is advised that this exercise is not intended to be either formal or exhaustive; it merely serves as an indication of the general functionality which might be supported.

Management policies are associated with managed objects. In the case of activity management, managed objects are defined by the components of activity definitions. For the sake of argument, this paper will adopt SDL as a means of specifying these elements, although other models such as the AAM would do equally well. SDL defines the following classes of managed object:

Managed objects in SDL
Activities
Roles
Message objects
Rules
Actions

A variety of management policies might be associated with each managed object, covering the entire range of timescales and functional areas described in section two.

Activities

Various management policies might be associated with an activity as a whole. The activity might be assigned a priority relative to other activities. Rules for constructing activity names might be specified. Policies for assigning a "coordinating agent" [4] (system or human) might be provided.

Roles

Role allocation/assignment is a key aspect of role management. Policies might be expressed in terms of lists of names or, where an X.500 Directory service is available, a named list or search criteria (e.g. this role may be allocated to anyone whose title is "programmer" and who is experienced in Fortran) [13]. Security policies might specify the authentication level associated with a role (e.g. must use a "strong" public key technique or may be unauthenticated) or may assign a security clearance level to the role (e.g. can access top-secret, classified or public information). Fault handling policies might specify "back-up" role players for a critical role.

Message objects

A range of management policies might be applied to message objects. Configuration policies might allocate message storage agents for specific classes of message. Security policies might specify encryption mechanisms or assign security levels for messages (e.g. a "vote" might be encrypted and secret). Performance policies might be concerned with recording the size and number of specific message classes.

Actions

Actions describe both the exchange and manipulation of message objects by various roles. Actions might provide a suitable hook on which to hang performance and productivity monitoring policies by measuring the number of exchanges or the number of objects created. For example, the performance of a "voting" activity might be measured in terms of the percentage of votes cast. The performance of a bulletin board by the frequency/volume of postings on a specific topic (thus supporting a manager in making strategic decisions such as when to remove a topic or split a congested topic into sub-topics). An alternative approach might be to set and measure time constraints on actions. Actions might also provide a hook for accounting policies (e.g. who pays for a specific exchange and how are they charged).

Rules

Rules are used to group and trigger actions. Policies might be specified for unconsumed or unexpected actions (the issue of defeasibility needs to be considered at this point). Performance might also be indicated in terms of unconsumed actions, possibly identifying bottlenecks or inefficiencies in the structure of the activity.

In summary, an SDL communication structure (or AAM Activity Template) defines the managed objects for a specific class of activity and therefore provides a framework on which management policies may be hung. This suggests that a formal notation for activity management, supporting policies such as those identified above, might be developed as an extension/annotation of an existing notation. It is crucial to be aware of the difference between global activity definitions and local management policies. This distinction is further explored in the following sections which proceed to discuss the management of multiple activities within distributed environments.

4. Distributed activity management

The view of activity management adopted by previous sections has been limited in two major respects:

- Activities were considered as if they existed in isolation from other Activities.
- No indication was given as to how management functionality might be realised within distributed computer systems.

In reality, activities do not occur in isolation. Activity management must therefore address the problems of multiple, concurrent activities sharing resources within a variety of environments. Furthermore, due to the increasingly distributed nature of cooperative working, applications based on activity models are likely to be implemented as distributed computer systems. Distribution will have a major impact on the management of activities. Consequently, this section considers the requirements of activity management within distributed systems. In particular, it describes the need for a distributed systems architecture for activity management to provide a framework for mapping between the functionality specified within the formal notation and underlying systems and services.

To offer a different perspective on this work, this section can also be seen as discussing the issue of inter-activity relationships. Previous work on activity models has briefly considered the relationships that may exist between activities and has indicated that these may take several forms. For example, parent/child relationships and relationships due to common roles. By considering the distributed management of multiple activities, this paper is defining another type of relationship, namely, "management relationships", between activities. Examples of management relationships might be:

- Relationships due to the sharing of resources.
- Relationships due to the relative priorities of activities.

The underlying model of activity management assumed by this paper is of various *environments* containing humans and resources which are dynamically allocated to activities according to management policies. The environment concept is of paramount importance to activity management and will be central to the development of a distributed management architecture.

4.1. Environments

An environment defines a space within which activity management occurs. In general, an environment represents some organisational structure such as a company, a department or a group and "contains" the humans and resources existing within that structure. The environment represents the boundary of administrative authority for the structure and therefore may also contain local management policies. Examples of environments might be, "The University of Nottingham", "The Big Computer Corp" or "The Sales Department within the Big Computer Corp". A key feature of environments is that they may be overlapping or even nested in terms of the both humans and resources they contain and the boundaries of administrative authority they define (e.g. departments within a company).

The concept of environments was developed within the AMIGO MHS+ project [6] and a more detailed discussion is presented elsewhere in these proceedings [14]. In addition, environments are closely related to *management domains* as described within network and systems management [9].

4.2. An architecture for distributed activity management

In order to realise the complex functionality of the activity management notation within distributed systems, a framework is required to express the relationships between managed objects, activities, policies and resources. This framework, called the *activity management architecture*, provides a mechanism for locating resources within potentially vast distributed systems and for constraining the effects of management policies, both in terms of organisational and system boundaries.

A range of distributed system architectures have been specified by previous CSCW research:

- The Amigo MHS+ project developed a distributed architecture for group communication systems including management protocols and operations.
- The Amigo Advanced project also specified a distributed architecture including a *General Activity Management Agent* with the ability to "instantiate, terminate, cancel and monitor Activity Instances".
- The Cosmos project specified a layered systems architecture containing the *Cosmos Information Service*, an activity information service supporting management functions for the configuration of activities [15].

This paper proposes that these architectures require further development in order to support the full range of activity management functionality identified in previous sections. The following paragraphs therefore outline an activity management architecture, based on a combination of the above approaches, which might provide a suitable foundation for this work.

Environments form the basic components of the activity management architecture. Each environment contains "local" Activities, resources, humans and management policies. In addition, each environment contains a local *Activity Management System* (AMS). The AMS receives management requests via a well defined set of operations and executes these according to the local management policies. Furthermore, the AMS maintains a local *Activity Management Information Base* representing the current state of known activities. It should be emphasised that the locus of effect of any management request is defined by the environment within which it is executed. For example, a request to allocate a role to a human with specific characteristics might involve a search of all humans in the local environment. However, humans outside of this environment would not be considered.

The key to distributed management is that environments might overlap or be nested, reflecting the relationships between organisations and systems. Furthermore, environments might span a range of geographical locations thus supporting distributed Activities. The activity management architecture now appears as a set of environments, potentially related in complex ways, each containing a local AMS and associated management policies.

A mechanism is required for the navigation of management functions to the correct AMS for a given activity. One approach to the navigation of management operations was identified within the Amigo MHS+ project where the global Directory service is used to store environment descriptions containing the names of their associated Activity Management Agents. Consequently, the target AMS for a given operation can be determined by inspecting the relevant environment description stored in the Directory. In addition, environment descriptions might identify current Activities and notable objects within the environment, thus providing a global activity information service. This use of the Directory to store environment descriptions is more fully described in the Amigo MHS+ final report [6].

4.3. Future development of the activity management architecture

The development of a detailed activity management architecture, complete with Activity Management Systems, environment descriptions and navigation mechanisms, requires further research. In particular, the following problems should be addressed:

- Specifying the functionality and protocols for an Activity Management System supporting the full functionality of the activity management notation.
- Describing the structure and contents of the Activity Management Information Base.
- Specifying mechanisms for navigating and executing management functions within a distributed framework of overlapping environments.
- Analysing the role of the Directory service in activity management and specifying the necessary Directory object classes and attributes to support this.

In summary, the development of an activity management architecture and associated Activity Management Systems is required in order to facilitate the mapping between the functionality defined by the activity management notation and the management of activities within distributed systems. This section has proposed a framework for this research by providing an outline of such an architecture.

5. Conclusions

Recent advances in the theory of Computer Supported Cooperative Work include the development of a number of modelling tools for representing patterns of group communication. In particular, the past year has witnessed the publication of several models and notations based on the concept of structured activities. Work is also progressing on the use of these models to design and configure CSCW applications based on emerging communications technologies.

Activity management refers to the support required for planning, administrating and controlling the execution of activities. This paper has identified activity management as a key requirement for the development and acceptance of CSCW applications based on activity models. Consequently, activity management is a high-priority research topic for the immediate future.

The paper has also outlined a framework for this research in terms of a set of management objectives and the identification of two key requirements of activity management. Activity management objectives were defined by range of timescales and functional areas. The timescales indicate that activity management is required for the short-term control of activities, for the mid-term administration of activities and for the long-term planning and development of new activities. The functional areas indicate that activity management is concerned with configuration and naming, security, accounting, performance and fault-handling issues.

The first key requirement identified by this paper is the need for a high-level notation for expressing the management requirements of specific activities. This notation would be used in conjunction with existing activity definitions to specify activity management policies which are local to a specific environment. Such a notation might be developed as an extension or annotation of current activity definition notations.

The second key requirement is for an *activity management architecture* supporting the implementation of these policies within distributed computer systems. Central to the specification of the architecture is the concept of an environment as an activity management domain. Each environment contains an Activity Management System, with an associated Activity Management Information Base, which executes management operations according to local management policies. The paper also discusses the support required for management within overlapping and nested environments.

Finally, the paper notes that research into activity management might draw on a wide variety of sources. These include the activity models themselves, models of network and distributed systems management, research into human factors, security mechanisms and existing communication standards.

References

[1] Pippa Hennessy, Steve Benford and John Bowers, "Modelling Group Communication Structures: An Analysis of Four European Projects", In SICON 89: Proceedings of the Singapore Conference on Networks, Singapore, 1989, IEEE Computer Press.

[2] Hugh T. Smith, "The Requirements for Group Communication Services", Research Into Networks and Distributed Applications - Proceedings of EUTECO '88, Vienna, 1988, North-Holland.

[3] John Bowers, John Churcher and Tim Roberts, "Structuring Computer-Mediated Communication in COSMOS", Research Into Networks and Distributed Applications - Proceedings of EUTECO '88, Vienna, 1988, North-Holland.

[4] Uta Pankoke-Babatz, "Computer Based Group Communication: the AMIGO activity model", Ellis Horwood, Chichester, England, 1989.

[5] Wilbur S and Young R E, "Management Aspects of a COSMOS", Cosmos document number 36.3, Cosmos Coordinator's Office, Queen Mary College, London, 1987.

[6] Hugh Smith, Julian Onions and Steve Benford, "Distributed Group Communication - The AMIGO Information Model", Ellis Horwood, Chichester, England, 1989.

[7] Kornel Terplan, "Communication Networks Management", Prentice-Hall, 1987.

[8] "Interim Report on the Cosmos Project", Cosmos report number 45.5 EXT, Cosmos Coordinator's Office, Queen Mary College, London, January 1988.

[9] ISO, "Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management Framework", ISO IS 7498-4, 1988.

[10] Starr Roxanne Hiltz, "Productivity Enhancement from Computer-Mediated Communication: A Systems Contingency Approach", Communications of the ACM, Volume 31, Number 12, pp. 1438-1454, December 1988

[11] Sead Muftic, "Security Mechanisms for Computer Networks", Ellis Horwood, Chichester, England, 1989.

[12] D.W. Davies and W.L. Price, "Security for Computer Networks", John Wiley & Sons, 1984.

[13] ISO and CCITT, "Information Processing Systems - Open Systems Interconnection - The Directory", ISO DIS 9594-1-8, CCITT X.500-X.521, 1988.

[14] P. Hennessy, "Information Domains in CSCW", Proceedings of the 1st European Conference on Computer Supported Cooperative Work, Gatwick, September 1989.

[15] R.B. Araujo, G.F. Coulouris, J.P Onions and H.T. Smith, "The Architecture of the Prototype COSMOS Messaging System", Research Into Networks and Distributed Applications - Proceedings of EUTECO '88, Vienna, 1988, North-Holland.