Group Management Services for Distributed Multimedia Applications

Silvester Namuye
United States International University

David Hutchison
Lancaster University

Andreas Mauthe
Lancaster University

Group Management Services for Distributed Multimedia Applications

Silvester Namuye, David Hutchison
Andreas Mauthe, Geoff Coulson
Computing Department
Lancaster University, UK
email: <namuye, dh, andreas, geoff>
@comp.lancs.ac.uk

1. Introduction

In today's computer network environments there is an increasing need to be able to connect named groups of people and/ or processes that need to collaborate to carry out some activity. We are concerned with the communication of multimedia information amongst distributed group members, and this places heavy demands on systems infrastructures and networks.

In current multimedia environments, the developers of group applications are forced explicitly to manage group membership, group characteristics, connection establishment, dynamic conference control and synchronisation **[Garcia 93]**. Ideally, however, applications developers should be left free to concentrate on the semantics of the application **[Trevor 95]**; if a class of application requires a certain functionality or quality of service, the lower layers should provide it **[Verissimo 92]**. Apart from freeing the user from the burden of providing these services, such lower layers can be specially optimised and thoroughly tested. In this paper, we provide an overview of the requirements of multimedia group applications and suggest an architecture for a set of Group Management Services (GMS) intended to provide generic system level support for this class of application. Implementation work on the GMS being carried out at Lancaster University is also described.

2. Group Communication Services

In the GCommS project **[Mauthe 94]** a basic set of services that satisfy the requirements of multimedia group applications have been identified. These services can be grouped as follows:

- *data transmission services* which are concerned with connection establishment and message transmission between a group of users, providing appropriate Quality of Service (QoS) for each individual participant;
- *synchronisation services* which provide real-time synchronisation between semantically related media streams, and ensure that the delivery of messages from different users is such that all users see the same ordering of events;
- *group management services* which deal with activities relating to group membership and group dynamics.

The purpose of this paper is to describe an implementation of the group management services within the GCommS project.

3. Group Management Services

3.1 Application Services

The group management services provide the following functionality:

- group creation
- group naming and addressing
- membership administration (i.e. joining, leaving, ...)
- group event notification (i.e. informing interested parties of changes in group state)
- group property management (i.e. authentication and rights and roles of members and groups).

3.2 Group Management Architecture

To provide the above functionality, we adopt an architecture consisting of a *group management* module, a *directory service* module, and a *data base* module. Applications interact with the group management module and with the directory service as depicted in the following figure. Note that all three of these architectural modules are logical entities and may themselves be distributed in implementation.

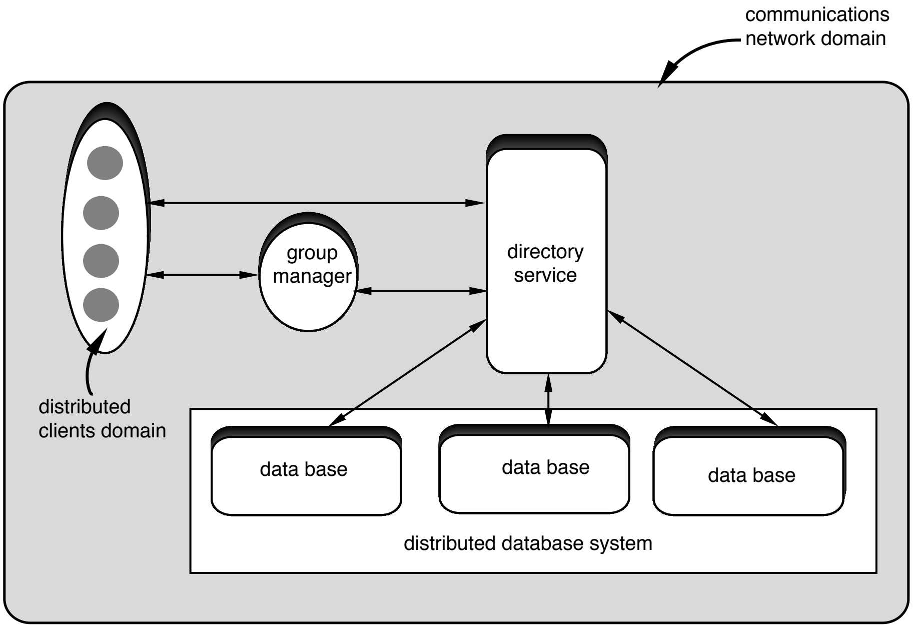

Figure 1: The Group Management Architecture

The group management module provides an interface to clients and uses the lower level services offered by the directory service and database system. It supports the creation of two types of group: *static* groups whose membership does not change, and *dynamic* groups whose membership can be changed according to associated authentication and group membership rules and properties. When a new group is created, the group management module contacts the directory service which provides it with a globally unique identifier. Newly created dynamic groups are provided (by the creator) with an authentication policy which constrains future group membership.

The membership administration, group event notification and group property management roles of the group management module are closely inter-related. The group management module provides *join* and *leave* operations which are subject to authentication according to the authentication policy specified at group creation time. In addition, clients may register with the group management module to be informed of *join/leave* etc. operations on particular groups.

Other facilities supported by the group management module include deletion of groups and permanent deletion of members, queries on current membership and queries on group characteristics. The group property management functions allow policies (e.g. the membership policy or visibility of groups) to be changed. It also allows *roles* to be associated with group members (e.g. a member may have the rights to create and destroy a group conference session or it may have no such rights).

The database system(s) store persistent group state and can be queried by the group management module. The directory service provides a database independent interface to the database system(s) and implements a global naming system. This allows the architecture to be configured with a variety of database technologies (perhaps multiple database technologies at the same time) without requiring changes to the group management module. The directory service can also be directly queried by clients if required.

4. Implementation of Group Management Services

The above architecture is currently being implemented on the Lancaster University campus network composed of the Computing Department network segment and an ATM based backbone. Our implementation strategy has been to treat the user, the group manager, the directory service, and the data base units as separate processes running on different hosts on our local area network. Our implementation environment is Unix/Linux with C++ and the 4.3 BSD Socket interface.

4.1 Group Management Module

The group management module offers the following interface:

create_group(gp_prop, member, opt_props);	// create a new group with given properties
destroy_group(group);	// destroy a group
remove_member(group, member);	// remove *member*
add_member(group, member_info);	// add *member*
request_group(group);	// return group property details
request_members(group);	// return membership list
req_group_list(domain);	// return all group names in domain
request_member(member);	// return all groups of which *member* is a member

Figure 2: Basic group management services

The implementation of the group management module consists of two basic object classes: the *group* class and the *member* class. A third object, the *policy* object, is yet to be implemented in this module. Complementary to the group management service, a client library is provided which abstracts over communication and gives access to remote clients.

4.2 Directory Service Module

The *directory service* module acts as a front end for the group management module to manipulate information in the data base. This involves the operations that will *insert*, *remove*, *list* and *search* for groups and membership attributes. These operations are accomplished by the following interface to the group management module:

add_group(group, features);	// insert *group*, *group* information, *group* features
destroy_group(group);	// destroy group
add_member(group, member_info);	// insert *member* and *member* info to given *group*
join_group(group, member_info);	// add existing group *member* to another *group*
leave_group(group, member);	// remove *member* from given group
request_group(group);	// return *group* property details
request_member(member);	// return all groups of which *member* is a member

It also provides the following interface for direct queries from clients:

request_member(member);	// return all groups of which *member* is a member
request_group_list(domain);	// return all group names in given *domain*

The directory service consists of an interface that has been implemented using C & ESQL (Embedded SQL) interfacing with the INGRES data base language. The data base contains the following information:

Group_info:	group_id, name, comments, max_members
Group_features:	state, type, dynamics, access, lifetime, membership_transparency
Member_info:	member_id, name, comments, host_name, host_address, port_nmuber
Member_features:	group_id, rights, roles.

The group and member *ids* are randomly generated and stored as long integers. The data base maps a group id onto multiple host_addresses of members. This reference is then used to communicate with group members on to point (not multicast at present) using UDP.

4.3 Application Scenario

An application may use the group management service in two ways. If an application is interested in registered groups and members information, it may contact the directory service directly by either group names and/or addresses and other related information.

If an application requires to establish communication with the group, it does so by contacting the communications manager to establish communication with the group using the group information obtained from the directory service. If further details are required by the communications manager, then the application again contacts the group management entity to supply the required information. It should be noted, however, that for complete communications set-up, other management entities are brought into play, e.g. the resource allocations manager.

4.4 Future Work

Currently our group management system runs on two processes: the group management module and the directory and database module. At the moment, the directory service interface is strongly coupled to the INGRES database. This needs to be made data base independent. Also there is need to investigate how the group management system will perform with a number of client requests. The configuration of the group management system is a pertinent issue: the need to structure group information in line with or parallel to conventional techniques such as Domain Name System, Global Name Service, or X.500.

5. Conclusion

In this paper we have proposed an architecture for group management services composed of a group management module, a directory service module and a database module. This architecture is part of a wider design that also incorporates services for communications support and synchronisation. The design as a whole will heavily rely on the support provided by the underlying communications system, which should provide low level group-addressing and multicasting facilities.

We are aware of some of the existing multicast support, but of particular interest is the envisaged IP Next Generation (IPng) which will have expanded addressing capabilities, with the inclusion of the "scope" field that limits the utilisation of multicast addresses. The scope field may identify that a datagram should be multicast to all on the same node as the sender; all on the same link as the sender; all on the same site as the sender; etc. These are transient addresses which are only meaningful within the defined scope and can provide a mechanism for scalability of groups.

REFERENCES

[Garcia 93] F. Garcia, "Continuous Media Transport and Orchestration Services", Ph.D Thesis, Lancaster University, 1993.

[Mauthe 94] A. Mauthe, D. Hutchison, G. Coulson and S. Namuye, "From Requirements to Services: Group Communication Support for Distributed Multimedia Sytsem", Second International Workshop on Advanced Telservices and High-Speed Communication Architectures (IWACA), Heidelberg, 1994, R. Steinmetz, Springer-Verlag, 1994, pp.266-277.

[Trevor 95] J. Trevor, "Infrastructure Support for CSCW", Ph.D Thesis, Lancaster University, 1995.

[Verissimo 92] P. Verissimo and L. Rodrigues, "Group Orientation: A Paradigm for Distributed Systems of the Nineties", Third Workshop on Future Trends of Distributed Computing Systems, Taipei, Taiwan, 1992, IEEE Computer Society, pp. 57-63.