4

DiSC - An Object Oriented Distributed Session Control Implemented with Java

E. Wedlund, C. Johnsson***
Ericsson Telecom AB
Switchlab, Dialoggatan 1, S-126 25 Stockholm Sweden
Phone: +46 8 719 85 96, +46 8 719 4903***
Fax: +46 8 719 66 77
etxelin@kk.ericsson.se, qtxcjo@kk.ericsson.se***

B. J. Olsson
Communicator Teleplan AB
Box 1310, S-171 25 Solna, Sweden
Phone: +46 8 7644500
Fax: +46 8 7644066
bengt.j.olsson@communicator.se

Abstract

Traditional session control is not a sufficient solution in a multimedia services network, it must be renewed in order to handle multimedia and multi-user sessions. In this paper we address this issue and present a proposal for a solution, which we call Distributed Session Control (DiSC). DiSC can be seen both as an overall solution for session control in a network, and simply as a tool for creating complex multimedia sessions. One of the strengths of DiSC is that it does not dictate a new type of network architecture, but will operate on those already existing.

Keywords

signalling, session control, network management, service provisioning

Management of Multimedia Networks and Services R. Boutaba & A. Hafid (Eds.)

1 INTRODUCTION

This paper describes an object oriented Distributed Session Control (DiSC) for multimedia services networks. The demands placed on a multimedia services network are that it should be flexible, in terms of introducing new services, and that it possibly should support QoS and security, and thereby also billing. Introducing new services can be simplified by moving session and network related issues to a dedicated session control function, such as DiSC, instead of having to deal with it in the applications.

In a multimedia services network, a session no longer consists of one user and one service (i.e. POTS), it can now consist of an unlimited number of users and services, which also can use different networks. The requirements that can be placed on this session control are:

- Complexity must not grow more than linearly with the number of users and services in a session.
- A user, or an application started by the user, should be able to use an appropriate QoS (if supported by the network).
- It should be possible to charge a user based on different session characteristics.
- It should be possible to use different networks for the same service.

Traditional telecommunication session control implies too much complexity to handle multi-user, multimedia sessions. DiSC solves the complexity problem by using an object oriented view of the session. Distribution of the session control also reduces complexity as well as scaling problems, since each user only keeps information that is relevant for that user.

2 DISTRIBUTED SESSION CONTROL

A project at Bellcore, called the EXPANSE project (Minzer 1991, 1993) recognised the problems above, and introduced an object oriented call model, and a concept of each user having a User Request Manager (URM), which handled negotiation for session establishment. This project was focused on the ISDN, but since we found it to be a useful way of handling the session control, not only for ISDN, we decided to use the same approach in our project.

The Distributed Session Control (DiSC) uses the same call model as the one in the EXPANSE project at Bellcore. In DiSC, each user also has its URM, which stores the object tree describing the session, and handles all negotiation with other URMs for creating and tearing down sessions. The URM can either be placed in the user's computer or in another place in the network. DiSC is implemented using Java Remote Method Invocations as communication between users and URMs. In

order to set up a session, the user sends a request to his URM stating which users and services he wants in the session. The URM will forward this request to the remote users' URMs. The user can also state which QoS he wants if that is supported by the network. Once a session is established, it is possible to add more users and services to the session, but it is also possible to start another session in parallel if desired. Once an agreement has been made about what services and users will join the session, the URMs will handle negotiation with the network layer to create the connections. If the user is talking directly to DiSC, the necessary applications will be initiated by DiSC as well. If it is an application using DiSC through an API, DiSC will deliver the connections via the API.

3 CONCLUSIONS

Currently, we have implemented DiSC as a simple conference tool with a few services available on an IP network. We have experienced that it is difficult to keep the flexibility of the LocalView when implementing it, since different services can have completely different attributes, and allow different types of session configurations. To solve this problem, rules can be stored in a database, where both the user and network administrator can make changes.

We have also come to the conclusion that the most likely future for DiSC is to be used as a middleware between an application and the network, and not as a tool itself, and therefore we are currently working on defining an Application Programmer Interface (API). This API should consist of methods for making requests, responding to requests, making changes in the rule database, etc.

A key feature of DiSC is that it hides the complexity of the session control from the user and application, which simplifies introduction of new services in a network. An application can be written for DiSC using an API, but already existing applications might also be used with DiSC. Another benefit is that DiSC itself does not require changes on lower levels in the network, since connections can be handled in a traditional way once the session has been created (separation of call and connection control). DiSC supports charging, since it has information about the session on a high level, which can be logged and used for e.g. service based charging.

DiSC is more light-weight than other approaches for multimedia services networks, e.g. TINA (Chapman et al. 1995), in that it only provides a solution for the session handling of the network, which means that it can be used on any existing network. DiSC is not defined for a specific network, which is the case for e.g. the ITU-T H.323 series. In fact, DiSC can interwork with these protocols for creation of connections on various networks, see figure 1.

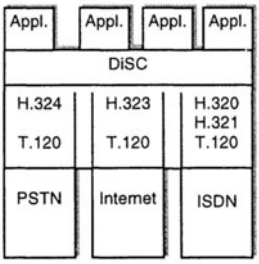

Figure 1. Network integration through DiSC and H.32x.

4 REFERENCES

Chapman, M. and Montesi, S. (1995) Overall Concepts and Principles of TINA, Version 1.0.

Minzer, S. (1991) A Signalling Protocol for Complex Multimedia Services. *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 9, 1383-94.

Minzer, S. (1993) EXPANSE Software for Distributed Call and Connection Control. *International Journal of Communication Systems*, 7, 149-60.

5 BIOGRAPHY

Elin Wedlund received the M. Sc. degree in Electrical Engineering from Lund Institute of Technology in Sweden in 1996. Her master thesis was on the subject of TCP and the ABR service in ATM. Since her graduation she has been working at SwitchLab, which is a lab within Ericsson's applied research organisation, where she studies networking issues.

Claes Jonsson started working at Ericsson in 1995, and has worked on the development of AXE switches before he came to SwitchLab. Alongside his work at Ericsson, he is also a student at the Royal Institute of Technology in Stockholm, where he studies Computer Science. At SwitchLab, he works with implementation issues.

Bengt J. Olsson received the M. Sc. degree in Engineering Physics 1985 and a Ph. D. in Atomic & Molecular Physics 1989 from the Royal Institute of Technology in Stockholm, Sweden, and did his Post doc. at the University of Wisconsin in 1990. He started working at Ericsson in 1991 with system design on SDH systems, and moved to Ellemtel Telecommunication Systems Laboratories in 1994 to become leader of the Network Studies group at SwitchLab. Since April 1997 B.J.O. works as a data/telecom consultant at Communicator Teleplan AB of Stockholm.