TOVE in Broadband Wireless Networks

Petteri Koponen, Pasi Nummisalo, Timo Pärnänen, Juhana Räsänen, Vesa-Matti Puro, Olli Martikainen

Telecommunications Software and Multimedia Laboratory
Helsinki University of Technology
{pkoponen, nummisal, parnanen, justus}@tcm.hut.fi,
Vesa-Matti.Puro@lut.fi, Olli.Martikainen@hut.fi

Abstract

To support a wide range of multimedia services future mobile network capabilities should allow the distribution of service functions between the customer terminals and the network nodes. Intelligent networks (IN) can be considered as the bridging technology between the customer based service management and the service execution in the networks. Considering the mobile and broadband technologies it is a natural question, how the concept of intelligence will develop. To satisfy broadband and multimedia service needs future mobile networks should support much more distributed operations. In the paper distributed TOVE (Transparent Object oriented Virtual Exchange) architecture is developed and applied in UMTS (Universal Mobile Telecommunication System). The presented TOVE architecture applies OVOPS++ (Object Virtual Operations System ++) software environment and distributes switching and different control functions using GSMP (General Switch management protocol) and INAP (Intelligent Network Application Part) protocols.

1 Introduction

With the introduction of multimedia and broadband mobile services there arises a need to control complex combinations of service components and resources. Broadband mobile networks can support interactive multimedia services with applications to electronic papers and publishing, digital interactive TV and audio, multimedia on demand services for banking, shopping, news and leisure. The technological requirements for these services are cost effective broadband transmission and access technologies, flexible computer based management and control of networks, switching and service applications and the support of mobility. When the broadband wireless access will be available in cost-effective form, then interactive business and consumer services based on video and multimedia will become possible.

In modern telephony the most promising computer control and management concept is the Intelligent Network (IN). It facilitates the development and management of new services by service providers. Intelligent networks can be considered as the bridging technology between the customer based service management and the service execution in the networks. It seems to be clear that the service applications will be distributed over terminals and different service control and provider nodes. An example of distributed intelligence can be found in the specifications of Universal Mobile Telecommunications Services [18]. A number of modifications and improvements to IN and call processing are necessary before future mobile telecommunication systems can be implemented. In this paper we present a distributed switching model supporting interactive multimedia services in broadband mobile networks.

2 TOVE Project

TOVE (Transparent Object oriented Virtual Exchange) project is a vertical research project that aims at developing a broadband network architecture that would support open service interfaces and flexible allocation of network control functions.

At the first phase, TOVE focuses on realizing middleware that defines distributed network control functions and interfaces used by different services and applications. The project is tied in with the integration of telecommunication and data networks. Due to relatively little systematic research in this area, prototypes are required in order to concretize the research. Pilot applications are implemented in parallel with the development of middleware, so that the control functions required by applications can be mapped onto the network control functions.

B-ISDN (Broadband Integrated Services Digital Network) is an attempt to set-up a single unified, global high-speed network for transmission of speech, data, image and video services [1]. The ITU-T (International Telecommunications Union - Telecommunication) is responsible for the standardization of B-ISDN. The ITU-T, previously called the CCITT (International Consultative Committee for Telecommunications and Telegraphy), selected ATM (Asynchronous Transfer Mode) as a transport mechanism for B-ISDN. Both the telecommunications industry and the computer industry have accepted ATM as a solution for public (B-ISDN) and private (e.g. Local Area Networks) networks. It is also expected that public and private networks are aligning to guarantee an overall worldwide compatibility [2].

In this environment, it is natural that TOVE architecture will be based on ATM technology. The first prototype of the project will be an open-interfaced ATM switch. The switch will be used as a platform for piloting different middleware solutions.

2.1 Virtual exchange

The ATM switch prototype will be based on FSR (Frame Synchronized Ring) [3], a new cost-effective switching architecture, developed and patented by the Technical Research Center of Finland. FSR is based on the ring topology and a simple yet efficient medium access control mechanism. FSR architecture is application-independent, but it is expected to perform well in switching ATM cells (cells are small fixed-length packets).

Current work in TOVE project is on implementing the protocol stack needed for ATM signaling. The signaling is used mainly in establishing and releasing ATM connections. An ATM switch must support the access signaling protocol between itself and the users and the interoffice signaling protocol with other switches. In TOVE project, the ITU-T standards are used (the ATM Forum is another organization working on ATM standards). The ITU-T standard Q.2931 [4] is the access signaling protocol across the UNI (User-to-Network Interface) and the interoffice signaling protocol across the NNI (Network-to-Network Interface) is B-ISUP (Broadband Integrated Services User Part) [5].

One of the main principles of TOVE architecture is to separate control from switching, i.e. to extract signaling functions from switching of cells. This approach is quite similar to one taken in DCAN (Distributed Control of ATM Networks) project [6] at the University of Cambridge. The major benefits gained from this approach are:

- The switches have to implement only a simple protocol that the switch controller can use to, e.g. establish, modify and release connections, configure switch ports and request and receive status information from the switch. This makes switches cheap and straightforward to implement. Also, if the switches implement a standard protocol, this will make the switching infrastructure vendor-independent, i.e. the switch controller need not to know the vendor or the type of the switch it controls.
- One switch controller can handle several switches. This is illustrated in figure 1. A controller workstation (switch controller) manages multiple switches. It holds information about e.g. the status of the switches, the topology of the "subnetwork" it manages, and the connections passing through the subnetwork. Signaling information from external sources (e.g. terminals or other switches) is switched through previously assigned channels to the controller workstation (and the other way around). The switches belonging to the subnetwork can not differ signaling information from plain data. The advantage of this transparency is that the switches do not have to be updated if the signaling protocol is modified or even completely replaced by a new one. Also, a device, which is connected to the subnetwork, can communicate with the switch controller by using an optimized, proprietary protocol.
- The architecture is scaleable. The number of switches controlled by one switch controller may be varied depending on the estimated signaling load. In case of distributable control functions, this could be taken even further: one switch could be managed by several switch controllers. In general, the signaling capacity may be enhanced by placing new switch controllers in suitable positions in the subnetwork (e.g. signaling bottlenecks). The architecture has been named as "virtual switch", because the external entities see the subnetwork as an individual switch, even if it physically consists of multiple connected switches.
- Interoperability with existing and future signaling standards. The virtual switch communicates with the external devices by using standard protocols. If these protocols need to be updated, this has to be done only in the switch controller(s).

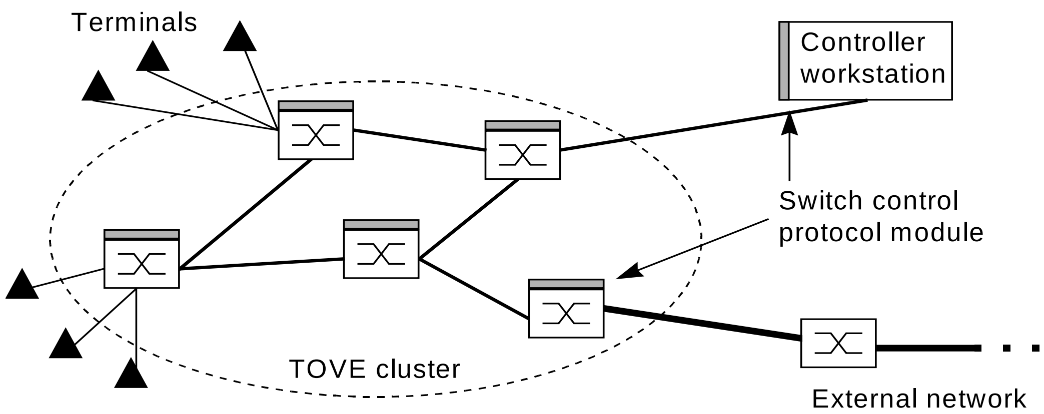

Figure 1 Architecture of the virtual exchange

An example of a suitable switch control protocol is Ipsilon’s GSMP (General Switch Management Protocol) [7], which has originally been developed for Ipsilon’s IP (Internet Protocol) switch [8]. Although GSMP supports only a limited number of control functions, the protocol has been considered suitable for TOVE architecture due to its extensibility and reasonably wide industrial acceptance (e.g. Digital Equipment Corporation and Hitachi have adopted GSMP in some of their switch models).

Separation of control and switching makes it possible to build a switch that has minimal hardware and software instead of general purpose workstation for signaling applications. This kind of simple switching units can be placed more freely in e.g. lamp posts, basements or roofs. This architecture also enables flexible addition of processing power by simply adding a new switch controller into a network bottleneck. Providing a standard interface (User-Network-Interface and Network-Network-Interface) for outgoing network interfaces makes TOVE cluster look externally like a "virtual" switch that is actually an independent subnetwork.

More flexibility is achieved by offering an open interface to the switching control and hooks for triggers that activate IN (Intelligent Network) functions. Provided that access control and autentication are properly taken care of, call and bearer control can be distributed to service functions, even to the user applications.

2.2 TOVE architecture in IN perspective

Another key issue in TOVE architecture is the separation of service control from call and connection control. This means that the operation and provision of new services is independent of the basic network functions. The approach taken is to follow IN (Intelligent Network) principles. ITU-T is responsible for the standardization of IN. The standardization process builds upon Capability Sets (CSs). The first Capability Set, CS-1, is already finished, and ITU-T is working on CS-2 and CS-3. In TOVE project, CS-2 has been adopted as a framework for IN-related functions, and also the development of CS-3 is monitored.

IN is an architectural concept for separating IN-specific additional functions (services) from the basic network functions (call and connection control) [9]. The goal of IN is the ability to introduce new services or change existing ones quickly without having to modify the basic network functions [10]. This is achieved by the concept of the DFP (Distributed Functional Plane) [11]. The CS-2 DFP is shown in figure 2. The picture is a simplification of the full DFP model (e.g. management-related functions and internetworking aspects have been left out for clarity).

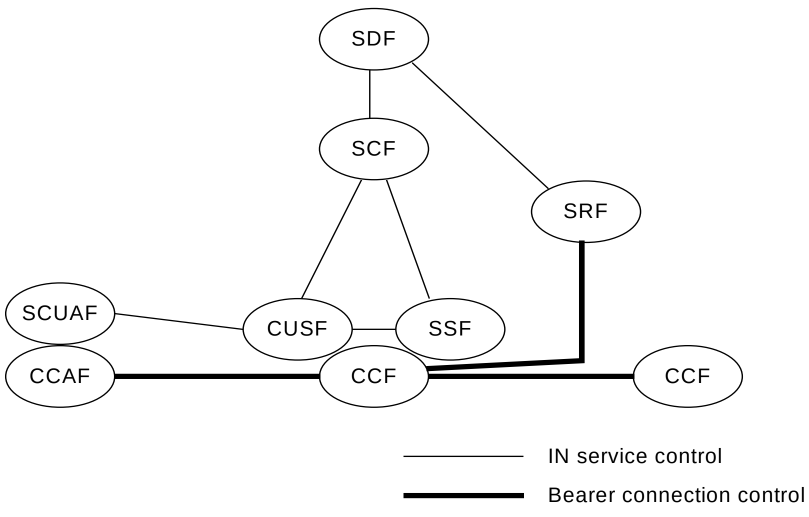

Figure 2 CS-2 DFP

The DFP consists of Functional Entities (FEs). FEs let the different network functions be described independently of their physical implementation or allocation in the network. One or more functional entities may reside in the same physical device, but according to IN principles, one FE can not be split between two devices [10]. The major functions of the FEs are the following [11]:

- The Call Control Agent Function (CCAF) is the interface between user and network call control functions.
- The Call Control Function (CCF) provides call/connection processing and control. It establishes, manipulates and releases calls. In addition to basic call processing, the CCF provides a triggering mechanism to access IN functionality.
- The Service Switching Function (SSF) is associated with the CCF and provides the set of functions required for interaction between the CCF and the SCF (Service Control Function). It is also involved in handling of non-call associated service.
- The Service Control Function (SCF) commands call control functions in the processing of service requests. One SCF can be associated with several SSF/CCF entities.
- The Service Data Function (SDF) contains customer and network data for real time access by the SCF in the execution of an IN provided service. The SDF can interact with other SDFs hiding the data location in the network. This can be used for data distribution transparency.
- The Specialized Resource Function (SRF) provides the specialized resources required for the execution of IN provided services, e.g. digit receivers and announcements.
- The Call Unrelated Service Function (CUSF) provides a set of call unrelated service functions required for out-channel interaction with the SCUAF (Service Control User Agent Function).
- The Service Control User Agent Function (SCUAF) is the interface between a user and the Non-Call Service Function (NCSF).

The basic idea of IN is that the SSF/CUSF/CCF needs not to be modified when introducing new services. The CCF includes the BCSM (Basic Call State Model), which is a high-level state machine description of the CCF activities required to establish and maintain communication paths for users. The required aspects of the BCSM are visible to the SCF via the SSF. These are the only aspects of the BCSM that are subject to standardization. The BCSM provides a framework for describing basic call and connection events that can lead to the invocation of IN service logic instances or should be reported to active service logic instances. This triggering mechanism allows the SCF to control the basic call processing and to take the actions necessary for service execution. [11]

The BCSM is not able to support multimedia and mobile services as required in the future. Due to this, an enhanced Call State Model will be developed in TOVE project. The demands for the call processing model supporting broadband multimedia and mobile services are analyzed in [12]. The processing model will be implemented in the switch controller software described earlier. Different models will be studied in order to find a suitable solution.

In typical configuration, SSF, CUSF and CCF form the SSP (Service Switching Point), which is e.g. a public exchange. The SCF can be integrated into the SSP, but usually it resides in its own physical entity, called the SCP (Service Control Point). In TOVE architecture, also the SSP can be divided into two entities. This is due to the separation of call and connection

control from switching. The SSP-A, which is the ATM switching fabric, is responsible for cell
switching and uses the switch control protocol to communicate with the SSP-B. The SSP-B contains the functionality of the normal SSP, except that it can control multiple SSP-A entities. This approach forms a three-level architecture, where the lowest level (switching) is relatively stable, whereas the highest level (services) is under constant modification. The intermediate level (control) may require changes to be made, when e.g. new signaling protocol is introduced or the call processing model needs to be updated to support new service classes.

In traditional IN architecture, communication between the call control and the service logic is implemented with INAP (IN Application Part) or its broadband equivalent Broadband INAP, which is currently under development. In TOVE, CORBA (Common Object Resource Broker Architecture) [13] has been considered as an alternative for the traditional solution, where service and call control entities exchange INAP (B-INAP) messages. CORBA makes service creation and implementation easier by providing a common language for defining interfaces, called IDL (Interface Definition Language), and by allowing objects, e.g. service applications, to be distributed transparently. Also, CORBA includes a standardized set of services (e.g. even services) that can be used

2.3 Implementation Aspects

Telecommunications software and protocols are becoming increasingly complex and at the same time they must be efficient, reliable and flexible. Also, software has to be developed in parallel with the emerging standards. OVOPS++ (Object Virtual Operation System) is an object-oriented extendible component framework for telecommunications software, which has been used in TOVE project to implement the signaling software for ATM. OVOPS++ is based an Conduits+ [14] and OVOPS [15]. The benefits of the OVOPS++ Framework are: the software parts become simpler and easier to maintain and evolve so time-to-market time is shorter, integration of additional functionality of new Capability Sets is easy and reuse of software for different standards (e.g. ITU-T and ATM forum) is possible [14].

2.4 Frameworks and Design Patterns

A framework is set of cooperating classes that make up a reusable design for a specific class of software. Framework will define the overall structure of a application: partitioning into classes and objects, the key responsibilities, how the classes and objects collaborate, and the thread of control.
A framework predefines these design parameters, so application designer can concentrate on the specifics of her/his application. The framework captures the design decisions that are common to its application domain. Framework emphasizes design reuse over code reuse. An added benefit comes when the framework is documented with the design patterns it uses. Design patterns flatten the learning curve making key elements of the framework's design more explicit. A typical framework contains several design patterns [16]. Design patterns are descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context. The design patterns used in Conduits+ and OVOPS++ are described in [16].

2.5 OVOPS and OVOPS++

OVOPS (Object Virtual Operations System) environment provides services to develop, interconnect and test object-oriented distributed applications. The OVOPS environment also provides a model, which improves the reuse of software and fasten the development cycle of applications. With OVOPS, implementations can be made largely independent of the operating system so that they can be ported to any system supported by OVOPS. A lot of experience from CVOPS (C-based Virtual Operating System) and OTSO (a Transport Service between two Objects) protocol environments have been used in the research of OVOPS. OVOPS++ uses core services of OVOPS, e.g. schedul-

ing, timers and frames, but the infrastructure is defined by the framework based on common object-oriented design patterns.

There are three ways to implement telecommunications software. With a classical design method and a language like C. This method is well known and efficient but the reusability and flexibility are poor. Code generation method uses CASE (Computer-Aided Software Engineering) tools to enable protocols to be defined with a specific language like SDL (Specification and Description Language). The CASE tool generates code of this language and every time the source code is changed it must be recompiled. During this process implementation specific details of protocol must be rewritten by hand over and over again . Object-oriented method has the advantages described in the preceding sections. Also compilers for O-O languages (in this case C++) are available for almost every platform, so we don't need to specify an another language to describe the behavior of protocol especially because we are using commonly accepted design patterns which can be implemented in C++.

The OVOPS++ Framework is made up of two sort of objects *conduits* and *messages*. A conduit is a software component with two distinct sides. A conduit can be connected on each of its sides. The conduit accepts messages from a neighbor conduit and delivers then to a conduit on the opposite side. There are four types of conduits. A *mux* can contain many neighbors on side B and it is used to multiplex messages arriving on this side and demultiplex messages arriving on side A. An *adapter* has no neighbor conduit on side B because it is used to interface the framework to some other software e.g. hardware driver. A protocol conduit is used to receive and transmit messages and provide commonly required facilities such as counters and timers. With state class protocol implements a Finite State Machine (FSM). A *factory* is used to create new instances of conduits at run time. Messages are transferred through the conduit hierarchy and they can be internal signals, primitives or PDUs (Protocol Data Units) [17].

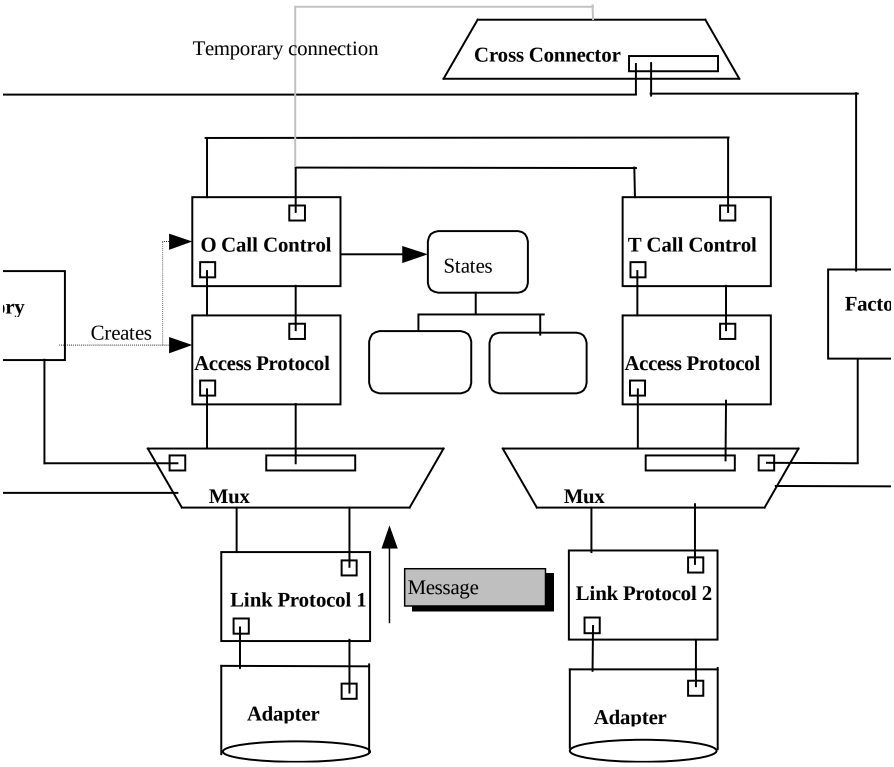

Figure 2 ATM switch with conduits

3 TOVE in UMTS

UMTS (Universal Mobile Telecommunication System) is a third-generation mobile telecommunications system, which aims to provide a wide range of communication services, ranging from speech communication and navigation services to video telephony and file transfer. UMTS will support both all the traditional mobile services and multimedia services up to 2 Mb/s.

Several services that are currently implemented by dedicated systems will be integrated into UMTS. These system include e.g. public mobile networks, cordless sets for domestic use, wireless LANs (Local Area Network), wireless local loop, satellite systems and paging networks. This implies that parts of the UMTS infrastructure will be installed and managed by public telecommunications operators and other parts will be in private ownership. Thus, UMTS will consist of zones of different size and ownership. UMTS architecture must support roaming, handovers and tariff scheme changes between these zones [18].

Most of the UMTS development has been done in RACE (Research and Technology Development in Advanced Communications Technologies in Europe) research programs. The development started within RACE I and the research work has continued within RACE II in several projects aiming to define UMTS system specifications. The standardization of UMTS is carried by ETSI (European Telecommunication Standards Institute). A similar project called IMT-2000 (International Mobile Telecommunication System 2000) is currently carried by ITU-T. UMTS and IMT-2000 are expected to be compatible so as to provide global roaming. Basic UMTS is scheduled to be available in 2002 and full UMTS services and systems for mass market services in 2005 [18].

2.1 Structure of the UMTS Network

The introduction of UMTS will be an evolutionary process. Rather than building an overlay network as in case of GSM (Global System for Mobile communications), parts of existing systems will be reused. ATM-based B-ISDN is going to be the dominating transmission technology on which the European information infrastructure will be built. Therefore, UMTS will be integrated with fixed B-ISDN networks. The integration requirement has led to a high level architecture, in which UMTS is divided into three different part as shown in figure 3 [19].

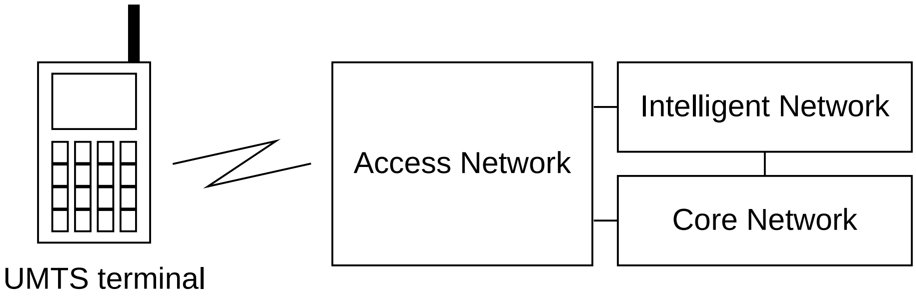

Figure 3 High level architecture of the UMTS network

The access network is responsible for most radio-related functions. It includes base stations and other equipment that enable interworking between the UMTS terminal and the fixed network platform.

The core network provides switching and transmission functions of UMTS. The core network is based on B-ISDN technology. The B-ISDN signaling needs to be enhanced, mainly to cater for handover execution and macro-diversity. Such enhancements are targeted for forthcoming B-ISDN CS-3 (Capability Set 3) that has a time frame similar to UMTS.

The mobile-specific control functions of UMTS are foreseen to be provided by IN functions based on IN CS-3. IN CS-2, which is currently under development, supports several mobility features, but CS-3 is expected to support fully functional UMTS network.

The target of the UMTS development is implement as much UMTS-specific functions as possible in the access and intelligent networks, so that the introduction of UMTS would require only minor changes to core network (B-ISDN) standards. However, due to the need for changes in B-ISDN standards, the future standardization of UMTS and B-ISDN will be done in cooperation, so that UMTS could be realized by reusing major fixed network components. [19]

2.2 Virtual exchange in UMTS

The

References

[1] Kyas, O., *ATM Networks*. International Thompson Computer Press, London, second edn.(1995).

[2] De Prycker, M., *Asynchronous Transfer Mode - Solution for Broadband ISDN*. TJ Press (Padstow) Ltd, Great Britain, third edn. (1996).

[3] Raatikainen, P., *Analysis and Implementation of a High-Speed Packet Switching Architecture - the Frame Synchronized Ring*. Doctorate thesis, Helsinki University of Technology, Espoo (June 1996).

[4] Q.2931 ++TODO++

[5] B-ISUP ++TODO++

[6] Roscoe, T., Rooney, S., *Overview of the DCAN project*. ++TODO++

[7] GSMP-viite

[8] IP switch -viite

[9] Q.1201

[10] Karttunen, T., Harju, O., Martikainen, O., "Introduction to intelligent networks." In Karttunen, T., Harju, O., Martikainen, O. (ed.), *Proceedings of the IFIP workshop on intelligent networks*, pp. 1-33, Lappeenranta, Finland (August 1994), Chapman & Hall, TJ Press.

[11] Q.1224

[12] Martikainen, O., Naoumov, V., Samouylov, K., *Call processing model for multimedia services*, IN'95 Conference, Copenhagen, 28 - 31.8.1995, 1 - 9.

[13] Viite CORBA:an

[14] Huni H., Johnson R., Engel R., A *Framework for Network Protocol Sofware.*

[15] Martikainen O. , Puro P., Sonninen J.: *OVOPS, Object Virtual Operations System for Distributed Application Development.* Proc. INDC 94, Fuchal, Madeira, Island, Protugal, April 18-21, 1994

[16] Gamma E., Helm R., Johnson R., Vlissides J., *Design Patterns, Elements of Object-Oriented software.* Addison-Wesley (1995).

[17] Engel R., Bieri T., Keller B., *Signalling in ATM Networks: Experience With an Object-Oriented Solution.*

[18] UMTS Task Force report

[19] Mitts, Håkan, *Use of INtelligent networks in UMTS.*