

A service engineering approach to inter-domain TMN system development

T. Tiropanis, D. Lewis, R. Shi

*Computer Science Department, University College London
Gower Street, London WC1E 6BT, United Kingdom*

Tel: +44 171 419 3687, +44 171 391 1327, +44 171 419 3249

Fax: +44 171 387 1397

E-mail: {ttiropan, dlewis, rongshi}@cs.ucl.ac.uk

A. Richter

GMD-Fokus

Hardenbergplatz 2, D-10623 Berlin, Germany

Tel: +49 30 25499 287, Fax: +49 30 25499 202

E-mail: richter@fokus.gmd.de

Abstract

The deployment of service management systems in a multi-service environment opens a whole area of issues. Requirements for openness and reusability can be satisfied by following a service engineering approach where the service management functionality is decomposed to reusable service components. Such an approach that was based on TMN technology is presented in this paper and conclusions are drawn about the potential of other available technologies with regard to service management.

Keywords

TMN, TINA, Service Management, Service Components, Distributed Processing

1 INTRODUCTION

In recent years much work in the area of distributed and telecommunication-based systems has been underpinned by an assumption that the resulting standards, interfaces

and products will have to operate in a global open market for telecommunication services and software. Such a market will be characterised by a more crowded and diverse arena of players than has been apparent in the traditional telecommunications industry structured around large state-owned enterprises. The increased competitiveness of this global market will produce ever increasing pressures to reduce costs and to increase service quality to the customer. Open interfaces have an important part to play in this both in reducing the burden on the service provider in dealing with multiple proprietary interfaces and in increasing choice to the customer by encouraging competition on the quality and cost of service rather than ties to specific interface technologies.

An area seen as being critical to cost reduction and customer satisfaction is that of management. This covers the management of the providers' resources to ensure their efficient use, the management of the delivery of the service to the customer and the integration of these activities to maximise the effectiveness of both.

The increasing management requirements on the network level, due to the use of the more complex ATM technology, and the demand for a number of elaborate service management areas such as subscription and accounting management, set an extra requirement for software reusability in the construction of network and service management systems. Principles of Object Orientation can find use in the decomposition of network and service management software to reusable components. Issues regarding the integration of these components are also considerable.

This paper presents work in the practical application of the TMN recommendations to the area of inter-domain service management by grouping part of the management functionality to reusable components. This work, undertaken in the RACE II PREPARE project [PREPARE], and later continued in the ACTS project Prospect, was conducted in an environment specifically designed to impose many of the requirements of an open service market on the development of a system of inter-working service and network management components; in this case managing multimedia tele-services over a broadband network. The requirement for openness justified the use of a service engineering approach. The next section outlines some of the standards that influenced this work. Section 3 describes the specific context in which this prototyping work was conducted. Section 4 details the design approach taken, the components used and the experiences gained before some conclusions are drawn.

2 CURRENT ARCHITECTURES

The most mature open architecture applicable to service management, and the basis chosen for the PREPARE project, is the ITU-T M.3000 series of recommendations on TMN. This standard has been widely accepted by the telecommunications industry in Europe and now increasingly in North America and Japan. The core architecture recommendation [M.3010] structures management functionality into a set of layers starting at the network element layer, through the network layer to the service layer and finally the business layer. TMN compliant standardisation efforts have led to a wide range of interfaces being developed at the network element level and more recently at

the network level. Open interfaces at the service level are only now starting to emerge, primarily intended for operation at the inter-domain interface identified in TMN, i.e. the X interface. The NM Forum is particularly active in this area through its SMART group. This group is addressing specific areas of management functionality identified in a common Business Process Model [NMF-BPM], such as trouble ticketing, service ordering and billing, where industry agreements on inter-work are required.

The process of defining open interfaces in TMN concentrates on the development of information models defined in terms of Managed Objects (MO) that can be accessed using management information retrieval protocols, e.g., SNMP and CMIP. The TMN Interface Specification Methodology [M.3020] provides guidelines for defining interfaces in terms of management interfaces that are decomposed into Management Service Components (MSCs) which are in turn decomposed into Management Functional Components (MFCs). However, in practice MSCs and MFCs are not often utilised as units of reusability for interface specifications, this being hampered further by a lack of a strict notation for these components. MOs are frequently reused between specifications, however without a clear mechanism for reusing MOs in groups forming MFCs the real value of this reuse tends to be more in the interface syntax of an MO than in the functional semantics it may offer. A further, large factor restricting widespread reuse of components in TMN is the current lack of a common API for management platforms to encourage the development of platform independent components implementations. This area is currently being addressed by the X-Open and the NM Forum [SPIRIT].

A different approach to component reusability has been taken by the Open Management Group (OMG) in developing its Common Object Request Broker Architecture (CORBA) [CORBA]. Drawing its membership from the wider data processing industry, as opposed to the telecommunication industry sector that influenced the development of TMN and the NMF, the OMG aims to apply to the area of distributed processing the object oriented techniques common in software engineering today, e.g. use of C++ classes as units of functional reuse. This is performed by providing mechanisms to support distribution transparencies, that have been identified as important in separating the design of a distributed system from the mechanisms providing, and the problems presented by distribution.

Significantly, the ODP [X901] recommendations go beyond outlining a technique defining interface specifications for distributed objects, and propose a general approach to defining distributed systems. This is based on the five ODP viewpoints: the enterprise viewpoint aiming largely at requirements capture; the technology viewpoint for defining the technological constraints placed on a system; the information viewpoint addressing the information to be processed in the system; the computational viewpoint addressing how the system is decomposed into computational objects and the engineering viewpoint describing how information and computational models are implemented as engineering objects, e.g. C++ classes.

Though the adoption of ODP techniques may encourage the development of a clear, object-oriented information model, the availability of object oriented platforms providing the appropriate distribution transparencies, e.g. a CORBA implementation, leads to the use of the computational objects of the computational viewpoint as the primary units of system decomposition and eventual reuse in the engineering model.

The reusability of such computational objects, and the corresponding engineering objects, is aided by the ODP engineering concept of clustering which allows for the grouping of engineering objects into larger units of functionality. This provides system designers with a wider and more continuous range of granularity in defining reusable components than it seems to be practised in TMN.

Though the ODP standards and the OMG architecture are applicable to any distributed processing problem, the Telecommunications Information Networking Architecture Consortium (TINA-C) have attempted to apply these principles to the needs of the telecommunications industry [TINA-018]. This has involved the adoption of ODP as the basis for their architectural approach resulting in the definition of an abstract Distributed Processing Environment (DPE) [TINA-005]. To provide continuity with existing telecommunications industry standards TMN principles have been adopted for the management aspects of this architecture while the influence of the IN standards [Q1204] is seen in the approach to functional decomposition and reuse. By bridging the gap between the emerging distributed processing industry and the well established telecommunications industry in this way, TINA provides a path for TMN-based management architectures to be integrated with the distributed systems techniques in a manner better suited to the requirements of the open service market. The following sections give an example of how this path could be followed based on some of the practical experiences gained from the PREPARE project.

3 OPEN SERVICE MARKET CONTEXT IN PREPARE

As stated in the previous section, the PREPARE project offered an opportunity to investigate the development of management services in an open service market context and in an environment that imposed requirements down to the implementation level. This consisted of an enterprise model imposing requirements to manage broadband networks and multimedia tele-services [Bjerring-94].

The enterprise model chosen consisted of a value chain of service providers. At the lowest level multiple public network operators (PuNOs) provide an ATM Virtual Path service over their respective network domains, and conspire to provide this service, spanning several domains, via a single management interface. This management interface is used by a Virtual Private Network (VPN) provider, in concert with the management of private network resources of ATM LANs owned by private network operators (PrNOs) to provide end to end management services connecting hosts on remote ATM LANs. This VPN service is in turn used by providers of multimedia tele-services, namely a multimedia conference (MMC) service and a multimedia mail (MMM) service in the provision of these tele-services to a customer organisation between internationally distributed ATM LAN sites.

The decomposition of management functionality between these various organisational stakeholders was performed initially at a very low level of granularity and in line with the TMN functional architecture. This involved identifying one TMN Operations System Function (OSF) for each stakeholder at the service level with additional OSFs, if required, at the network level and the network element level via Q-adaptor functions. The aim was to implement this TMN architecture as a set of inter-

operating Operations Systems (OSs) mapped simply to the OSFs. Figure 1 shows schematically how these OSs were distributed between organisational stakeholders and mapped over a broadband demonstration network, spanning London, Berlin and Copenhagen, and consisting of ATM LANs and ATM cross-connects (XCs). The X and Q3 interfaces developed between these OS are also identified.

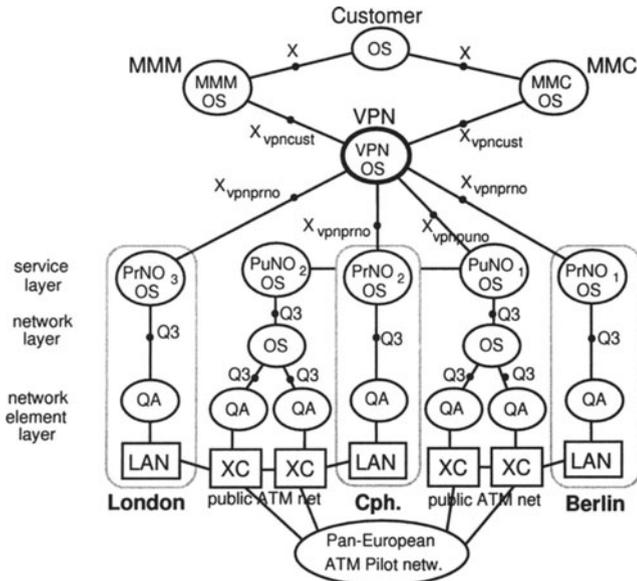


Figure 1 Overall TMN Physical Architecture for PREPARE.

4 APPLYING SERVICE ENGINEERING PRINCIPLES TO TMN OSF DESIGN

The diversity of the offered services in PREPARE added a requirement for openness and reusability to the design and specification of the reusable service management components. The need that the development of these components had to be shared between different project partner organisations emphasised this requirement.

4.1 Decomposing services into service components

Several OSs were derived from the enterprise specification in PREPARE. For the service management support for the services, corresponding service operation systems (S_OS) were defined. The communication between S_OSs in different domains was accomplished over the TMN X interface. A common information model in GDMO and ASN.1 specified the information to be exchanged over the X interface. A set of event-trace diagrams defined the X interface interactions between the S_OSs for a set of different service management scenarios.

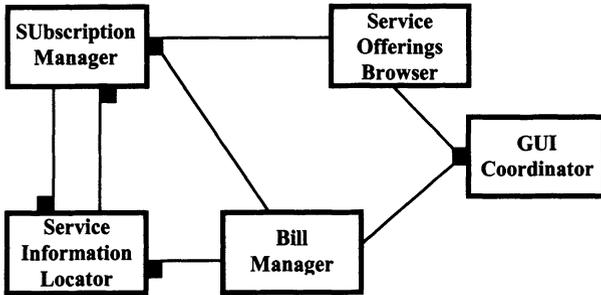


Figure 2 Relationships diagram for the generic Service Components.

The functionality of the S_OSs was decomposed to Service Components (SC). These SCs were categorised to *generic SCs* and *service specific SCs*. All the generic SCs are the reusable service management modules. The service specific SCs dealt with requirements for service specific functionality. The integration of management and service specific components is an idea borrowed from TINA.

The following generic SCs were identified for supporting the management of the services in PREPARE:

- The *Service Offerings Browser* (SOB) SC offers functionality for browsing through details for services on offer. The details are stored as entries on the X.500 directory.
- The *Subscription Manager* (SUM) SC implements the functionality needed for managing existing subscriptions to services. It employs a unique identifier for each service subscription and which is also used for de-multiplexing subscription specific service requests exchanged between SCs.
- The *Service Information Locator* (SIL) SC provides functions for locating service management agents on other customer or provider domains. For a given subscription ID, this SC can return the address of the customer or provider service management agent by using the combination of X.500 and X.700 described in section 4.3.2.
- The *Bill Manager* (BM) SC offers the utilities required for collecting, browsing, and paying of bills between customers and providers for a specific subscription.
- The *Graphical User Interface coordinator* (GUI_C) SC implements the functions for integrating the GUIs of the several service components installed and running on the same workstation function within a common front-end.

We mapped each SC to one Computational Object (CO) [TINA-HV][X901]. The relationships between the COs are shown in Figure 2 using TINA notation.

The GUI_C SC was used by both the SOB SC and the BM SC in order to register their graphical user interfaces within the same front. The SUM SC was used by the SOB SC in order for the latter to perform on-line subscription to a service on behalf of a user, and it was also used by the SIL SC in order to authorise the requester of the location of a service management agent. The SIL SC was used by the SUM SC in order

to find the contact agent of a service provider when a subscription was requested by a user, and it was also used by the BM SC when sending or paying bills.

4.2 Building-blocks and contracts

SCs can be grouped together according to the dependencies between them. Grouping of SCs allows for their better management, i.e., they can be treated as an entity that can be moved or copied to other locations and plug into other applications by offering a combined set of interfaces to them. The notion of a building block (BB) in TINA is that of a group of computational objects that has these properties. In this document the term *contract* is used for the interfaces offered by building blocks.

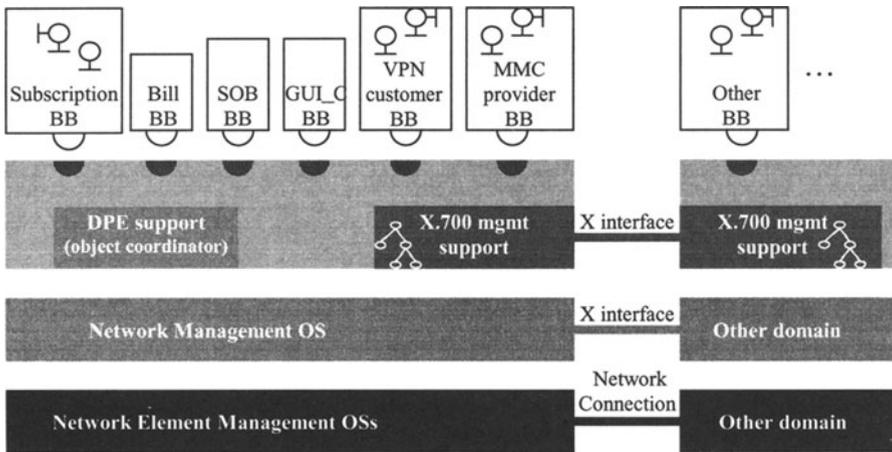


Figure 3 Structure of TMN S_OSs with building blocks and contracts.

The relationships between the SUM and the SIL in Figure 2 indicate the strong dependencies between them. For this reason they were put together in the Subscription BB. The other SCs were organised into separate BBs. The Bill BB includes the BM SC, the service offering browser (SOB) BB includes the SOB SC and the GUI coordinator (GUI_C) BB includes the GUI_C SC.

Most of the above BBs include an extra *coordinator* SC that is responsible for the initialisation, termination, activation and deactivation of the contracts of the BB. This component is also responsible for other generic building block management functions.

The service specific components were organised into separate BBs. There are two types of BBs for each service: the customer BB and the provider BB, e.g., for the VPN service, there are the VPN service customer BB and the VPN service provider BB. The same applies for the MMC and MMM services.

A picture of all the BBs organised on the site of the MMC provider is given in Figure 3. This combination of BBs comprises the VPN customer OS and the MMC provider OS and shows the case of being the MMC provider and a VPN customer at the same time. The communication between the BBs was accomplished over a distributed processing support infrastructure (DPE support) that was developed in PREPARE,

which resembles the TINA DPE. The inter-domain communication which was performed over the TMN X interface was supported by an X.700 management support module which was integrated with the DPE. Both the DPE support and the X.700 management support entities are explained in detail in section 4.3.

4.3 Key service components

Since no CORBA or TINA compatible platforms were used in PREPARE, two custom service components were developed. The object coordinator was to support TINA DPE-like functionality. The X.700 management object was to support an X.700 management gateway to the DPE. It was used as a special DPE service by other service components that utilised the TMN X interface. The TMN X interface was used only for inter-domain transactions in PREPARE as shown in Figure 3.

The object coordinator

Although in Figure 3 the object coordinator appears to be one layer underneath the other service components, it can also be considered as a special service component. It provides location transparency to all the contracts on a global level, by means of a unique global name it assigns to each of them.

The object coordinator offers an interface, with a well known reference, for the other contracts to use. It consists of the following methods:

- *Register*. This method is invoked by a contract of a BB when it registers with the object coordinator.
- *Deregister*. This method is invoked by a contract when it wishes to de-register.
- *Reregister*. This method is invoked by a contract when it wishes to amend its registration details, such as location. This can provide object mobility support that can be used for personal or terminal mobility.
- *Call*. This method is invoked by a contract when it wishes to call a method provided by another contract.

For scalability reasons, many object coordinators can be instantiated across the network. They can set up connections between them and via special interfaces they can:

- *Locate* contracts registered with any other object coordinator.
- *Ensure the uniqueness* of a global name when a contract registers.
- *Invoke* methods on contracts registered with any of the object coordinators.

On a high level, an object coordinator resembles a DPE or an Object Request Broker (ORB), while the communication between different object coordinators resembles the concept of the Kernel Transport Network (KTN) in TINA. In PREPARE, the object coordinator was used only for intra-domain access. All the transactions between different domains were made over the TMN X interface. Therefore, the inter-

coordinator communication was restricted to the boundaries of each administrative domain.

The object coordinator proved to be a core component for applying and trying out the TINA principles on our design. Although we cannot claim that it substituted the functionality a DPE can provide, it fulfilled our requirements for a DPE support component.

The X.700 management object

The X.700 management object is a component that was developed as a DPE service. It offers an X.700 management API to other components which can handle X.700 operations, actions and event reports. In PREPARE, it was used by contracts that utilised the TMN X interface for inter-domain management interactions.

The implementation of this component that was used in PREPARE, offers access to the X.500 directory as well. It is based on the integration of the X.500 directory and X.700 management which provides location transparency for X.700 agents that have an entry in the X.500 directory [PREPARE]. Each entry includes the address of the corresponding X.700 agent. Due to similar naming conventions for X.500 data objects (directory objects) and X.700 data objects (managed objects), the concatenation of these two provides a unique name for each X.700 data object regardless of which agent is hosting it.

4.4 Technology used

The object coordinator and all the service components were implemented in Tcl which satisfied requirements for flexibility and simplicity when developing service management logic. Tcl is an interpreted scripting language. As a high level language it can be used by service managers who are not necessarily experienced programmers. Since it is embeddable into C or C++ code, it is also possible to reuse already existing software and integrate it with Tcl code.

The Tcl Object Coordinator (TOC) is the implementation of the object coordinator component. It is based on Tcl (version 7.4) [TCL-TK], the object-oriented Tcl extension ObjectTcl (version b1.1) [OTcl], and the distributed programming Tcl extension Tcl-DP (version 3.3b1) [Tcl-DP].

There are two implementations of the X.700 management object. One is based on tcl-rmib which is part of the OSIMIS 4.0 management platform [OSIMIS][TCL-RMIB]. The other one is based on tcl-idmis which is part of the ANDROMEDA [Dittrich-95d] platform, which, in turn, is based on OSIMIS and ISODE.

Both, tcl-rmib and tcl-idmis, offer a Tcl-based X.700 management interface, where management operations are invoked by Tcl commands. The parameters passed to and the results returned from the X.700 management object are strings. To support the asynchronous reception of event reports they are mapped to Tcl call-backs. Tcl-idmis integrates X.500 and X.700 under the same API and therefore offers location transparency for X.700 management agents.

5 CONCLUSIONS

Though the PREPARE open service market context is a fairly limited example of potential enterprise situations that could evolve, it gives us some insight into the problems that a management system designer may face while working in this area. Such a context places new requirements on management systems as they encompass the integration of service and network management. One key requirement is the need for modularity of management systems at an engineering level. This is seen as essential in order to obtain the level of implementation reuse and the resulting fast time to market that competitive pressures and changing customer requirements will demand in the open service market. TMN recommendations offer the possibility for defining interfaces in a modular way, e.g. by using MFCs. However the mapping of modularity in interface specification to modularity in implementable components is outside the scope of TMN. This coupled with the slow emergence of an effective platform independent TMN API has meant that there is currently no commonly practised approach of modular TMN system design.

ODP, with its five view-points, supports implicitly the mapping modularity in the specification of functional interfaces to modularity in an engineering model. ODP has been taken by the TINA Consortium and applied to the telecommunications arena, though to date practical application of this architecture has been limited to a few prototypes. The work presented in this paper is an example of such a prototype, where some of the aspects from the TINA overall architecture have been applied to a home-grown DPE offering location transparency between engineering objects. By integrating this DPE with an existing TMN platform, the engineering aspects of applying ODP and TINA principles to TMN have been investigated to some extent. This is an important area, since even if TINA DPE or CORBA implementations are taken up increasingly for management applications there is a large investment in TMN-based management interface specification and an growing base of installed TMN agents and applications that will have to be accommodated by the new platforms.

The ACTS project, Prospect, has been building on the PREPARE and other related work in setting up further management networks for integrated multi-domain services in which an common engineering approach can be exercised. In PREPARE, the inter-domain interactions were supported over the TMN X interface and only the intra-domain interactions were done over a distributed processing platform. The use of CORBA platforms for uniform communication among components across domains in Prospect [Lewis-97] was considered beneficial to the approach followed in PREPARE. While TMN is still used on the network management layer in Prospect, due to the availability of such systems and to the previous investment in this area, a CORBA-TMN gateway is being developed to bridge the gap between CORBA-based service management components and the TMN-based network management components.

6 ACKNOWLEDGEMENTS

All partners in the PREPARE project and in particular Sven Krause, Ingo Busse and others at GMD-Fokus, Lennart Bjerring of L.M. Ericsson, as well as George Pavlou

and David Griffin of UCL. This work was conducted under the partial funding of the EU through the RACE II PREPARE project (contract R2004) and of the ACTS Prospect project (contract AC052). The views expressed in this document do not necessarily reflect those of the PREPARE consortium or of the Prospect consortium.

7 REFERENCES

- [Bjerring-94] Bjerring, L.H., Schneider, J.M., End-to-end Service Management with Multiple Providers, Proceedings of the 2nd International Conference on Intelligence in Broadband Services and Networks, Aachen, Germany, September 1994, pp 193-206, Springer-Verlag, 1994.
- [CORBA] *The Common Object Request Broker: Architecture and Specification*, OMG Document Number 92.12.1, Revision 1.1, Object Management Group, 1992.
- [Dittrich-95d] A. Dittrich, *The ANDROMEDA Platform: an object-oriented development and run-time environment for management services*, Proceedings of the 6th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM '95), 1995.
- [Lewis-97] D. Lewis, T. Tiropanis, C. Redmond, V. Wade, *Inter-domain Integration of Service and Service Management*, Proceedings of IS&N '97 Conference in Como, Italy, May 1997.
- [M.3010] *Principles for a Telecommunications Management Network*, ITU-T Recommendation M.3010, 1992.
- [M.3020] *TMN Interface Specification Methodology*, ITU-T Draft Revised Recommendation M.3020, 1994.
- [NMF-BPM] *A Service Management Business Process Model*, Network Management Forum, Morristown, NJ, 1995.
- [OTcl] *Object Tcl (version b1.1)*, <http://www.x.co.uk/devt/ObjectTcl/>, IXI Limited, 1995.
- [OSIMIS] G. Pavlou, G. Knight, K. McCarthy and S. Bhatti, *The OSIMIS Platform: Making OSI Management Simple*, in Integrated Network Management IV, ed. A.S. Sethi, Y. Raynaud and F. Faure-Vincent, pp. 480-493, Chapman & Hall, 1995.
- [PREPARE] J. Hall (Ed.), *Management of Telecommunication Systems and Services*, Springer-Verlag Berlin in Heidelberg, 1996.
- [Q1204] *Intelligent Network Distribution Functional Plane Architecture*, ITU-T Recommendation Q.1204, 1993.
- [SPIRIT] *X/Open Consortium Specification. SPIRIT Platform Blueprint*, SPIRIT Issue 2.0, Network Management Forum, Morristown, NJ, 1994.
- [TCL-TK] John K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley Publishing Company, 1994.
- [TCL-RMIB] T. Tin, G. Pavlou, *Tcl-MCMIS: Interpreted Management Access Facilities*, Proceedings of the Sixth IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM '95), Ottawa, Canada, October 1995.

- [Tcl-DP] B. C. Smith, L. A. Rowe, *Tcl Distributed Programming (Tcl-DP) (version 3.3beta1)*, <ftp://mm-ftp.cs.berkeley.edu/pub/multimedia/Tcl-DP>, June 1995.
- [TINA-005] Graubmann, P. and Mercoureff, N., *Engineering Modelling Concepts (DPE Architecture)*, TINA Baseline document TB_NS.005_2.0_94, December 1994.
- [TINA-018] Chapman, M. and Montesi, S., *Overall Concepts and Principles of TINA*, TINA Baseline document TB_MDC.018_1.0_94, February 1994.
- [TINA-HV] Natarajan, N., Dupuy, F., Singer, N., Christensen, H., *Computational Modelling Concepts*, TINA Baseline document TB_A2.HC.012_1.2_94, February 1995.
- [X901] *Information technology - Open Distributed Processing - Reference Model - Part 1: Overview*, ITU_T Draft Recommendation X.901/ISO/IEC Draft International Standard 10746-1, 1995.

8 BIOGRAPHIES

Thanassis Tiropanis graduated from the department of Computer Engineering and Informatics at University of Patras, Greece, in 1993. Since 1994, he has been working as a Research Fellow at UCL. Thanassis was involved in the European projects PREPARE and Prospect investigating the deployment of service management systems for the open services market. He has lead a team that investigated the introduction of mobility to the Prospect multi-service environment. Architectural aspects of mobile services is an area of particular interest on which he is doing a Ph.D.

David Lewis graduated in Electronic Engineering at the University of Southampton in 1987 and received an M.Sc. in Computer Science from UCL in 1990, since when he has worked as a research fellow. He has worked primarily on the EU funded projects PREPARE and Prospect, in which he has been responsible for planning high speed international test-bed networks and for leading teams developing integrated, multi-domain service management systems. He is also working on a Ph.D., researching a service management development architecture for the open services market.

Rong Shi graduated from the department of Computer Science at Tsinghua University, Beijing, China, in 1987 and received an M.Sc. at the same department, in 1989. Since 1994, she has been working as a Research Fellow at UCL. Rong Shi was involved in the European ESPRIT MIDAS project and RACE II PREPARE project. Now, she is working on the ACTS MISA project investigating the inter-domain routing algorithm for the SDH and ATM integrated networks. She has lead one activity in the MISA project on the system high level specification.

Alexander Richter received his diploma in Computer Science from the Technical University Berlin, Germany. During his studies he joined the Research Center for Open Communication Systems (FOKUS) of GMD in Berlin where he is still working in the Department for Management in Open System (MINOS).

Mr. Richter worked for the RACE II project PREPARE. After this project was finished he has been working for the ACTS Project PROSPECT where his scope is the management of services in the open service market. In this area the mobility aspects of service usage and provision are of interest.