Architectural Considerations for Managing Mobility

by

Athanassios (Thanassis) Tiropanis

A dissertation submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy
of the
University of London

Department of Computer Science
University College London
University of London

October 2000

Abstract

Ever since the first communication systems were deployed, requirements for mobility have been one of the most important causes of architectural changes. For instance, it has taken a long period of research and standardisation to arrive at the deployment of mobile telephony, the architecture of which is very different from that of the traditional/fixed telephony. Similarly, in the Internet, the proposed architecture of Mobile-IP adopts different methods of IP address allocation and routing, among other changes.

On the other hand, the mobile system architectures introduced by the different communities, e.g. telephony, Internet, data management, nomadic computing, and wireless communications, all differ significantly. They provide different kinds of mobility, they adopt different approaches to managing mobility and each of them has certain architectural problems. For example, the problem of triangular routing in Mobile-IP is an important drawback of its architecture.

Existing mobile system architectures are restricted by their legacy. Also, their scope is confined to particular aspects of mobility. For these two reasons, these architectures are not generic, and they cannot be extended to support additional kinds of mobility.

In this thesis, we firstly discuss the *problems of existing mobile system architectures*. Then, we abstract mobile systems to a generic level and we define the common denominator of mobile systems in a *framework of terms*. Using the terminology of this framework we examine the problem of managing mobility on an abstract level and we arrive at *architectural requirements* for mobile systems deployment, which are incorporated into the framework.

These requirements for mobility are investigated from two different ODP viewpoints: the *information viewpoint* and the *computational viewpoint*. From the information viewpoint we justify the necessity for *unique IDs* for identifying mobile entities and we discuss the nature and the scope of these unique IDs. From the computational viewpoint we outline a

number of entities that a system should feature in order to support mobility. The computational viewpoint requirements are specified following a *design pattern* template.

The hypothesis that motivated this thesis is that a technology independent framework for mobile systems is feasible. Also, that it can be generic enough to represent a variety of mobile systems and that at the same time it can be of practical value. Finally, there is a claim that such a generic and technology independent framework for mobile systems can help to consider many aspects of mobility and therefore assist to building coherent systems that avoid pitfalls common in existing mobile system architectures.

In order to investigate the validity of this hypothesis a framework of terms for mobile systems is specified. Using the terminology that this framework provides architectural requirements for mobility are investigated from the information and the computational ODP viewpoints. To test their generality, the terms of this framework are mapped to a range of real systems such as Mobile IP and GSM. A system for the novel area of *role mobility* that respected the specified requirements of the framework was deployed in a TINA environment in order to investigate the practical value of this framework and the requirements that are expressed in its terms. The design and implementation of the role mobility system is presented in this thesis. Additional to this, this framework was used to investigate some well-known problems of existing mobile systems like GSM and Mobile IP. In this way, the practical value of the framework and the architectural requirements that where expressed in its terms are validated further and the claim that this framework can assist in building coherent systems is substantiated.

This thesis, therefore, is intended to provide an insight into the problem of mobility and further directions to carry this research forward are suggested.

Acknowledgements

I would like to thank my fiancée Monica Assetta Binda, without whose constant support, encouragement and tolerance this thesis would not have been completed. Monica, this thesis is dedicated to you.

My supervisors Jon Crowcroft and Stephen Hailes for their useful feedback and continuous support through all the stages of preparing this work. Thank you for trusting this area to me and for believing in me.

The people who have funded my working at UCL and provided an excellent environment to work in: David Lewis, George Pavlou and Peter Kirstein. Also, David Lewis and Chris Malbon for providing me with useful comments on drafts of this thesis.

The team of the European ACTS project Prospect (contract number AC052) for providing the environment to develop and test many of the ideas in this thesis. In particular, many thanks to Lennart Bjerring, Alexander Richter and Sven Krause for the endless discussions on the different kinds of mobility. Many thanks to Chris Malbon, Hervé Karp, Nelly Cardinale and Catherine Burvelle for their useful feedback and for implementing the role management component. Many thanks to the PCS group of Prospect and the entire Prospect team for ensuring the successful demonstration of the role management component.

I would like to thank my parents and my family for encouraging my starting and completing this thesis.

Many thanks to friends with whom I have worked or who provided me with moral support through this period. Particularly to Kevin McCarthy, Thurain Tin, Alina DaCruz, David Griffin, Saleem Bhatti, Isidor Kouvelas, Giannis Koufakis, Natascha Vassila, Yiorgos Chrysanthou, Panos Gevros and Theodor Pagtzis.

Finally, I feel indebted to everybody who, from the position of a teacher or not, has ever taught me anything about science, history, art, "life, the universe and everything"...

Notes

Throughout this thesis the author chose to employ the first person in plural to address the reader. However, this does not imply collaborative work. Expressions such as "our hypothesis", "our thesis", "our approach", "we investigated" or "we claim" should not be perceived as indications of collaborative work. They should only be seen as a choice of style of the author. The work presented in all the chapters of this thesis is the author's, including the design of the role management component.

The content of Appendix A is a summary of the description of the CORBA design pattern template by Thomas J. Mowbray and Raphael C. Malveau based on their book "CORBA Design Patterns".

The only part of this document that presents collaborative work is section 5.2 which describes the implementation of the role management component in the European ACTS project Prospect (project number AC052). The implementation and demonstration of this component was carried out by the Prospect team as acknowledged in the acknowledgements section.

Table of Contents

Page

1. Introduction 13
1.1 Current standards and research 16
1.1.1 Internet 16
1.1.2 Telecom 16
1.1.3 Other research 19
1.2 The problem area 21
1.3 Hypothesis 26
1.4 Thesis structure and approach 27
1.4.1 Overall approach 27
1.4.2 Concepts and tools 28
1.4.3 Detailed approach 31
1.5 Outline of dissertation 34
2. Current mobile system architectures and related work 36
2.1 Mobile IP 37
2.1.1 Criticism 39
2.2 GSM 42
2.2.1 Criticism 46
2.3 Other Telecom standards 49
2.4 Related work 49
2.4.1 ODP and multimedia systems 49
2.4.2 The OSI reference model 50
2.4.3 Addressing and locating network entities 50
2.4.4 Formal models for mobility 50
2.4.5 Other related work 56
3. A Framework of Terms (FoT) for Mobile Systems 57
3.1 Communications network 58
3.2 Network entity (NE) 61

3.3 Medium 62
3.4 Communications network topology (CNT) 64
3.5 Communications path (CP) 64
3.6 Message exchange (exchange) 66
3.7 Entity location (location) 67
3.7.1 Local network entity (local entity) 67
3.7.2 Remote network entity (remote entity) 67
3.8 Fixed network entity (FE) 69
3.9 Mobile network entity (ME) 71
3.9.1 Location space 72
3.10 A definition of mobility 72
3.10.1 Relative speed of a mobile entity 73
3.11 Mobile entity classes 73
3.11.1 Mobile entity class (MEC) 74
3.11.2 Network entity class (NEC) 75
3.11.3 Complete Entity Class (CEC) 75
3.12 Entity identifiers (ID) 77
3.12.1 Fixed entity identification (FID) 77
3.13 Relating the FoT to existing models for mobility 82
3.13.1 Relating the FoT to the π-calculus 82
3.13.2 Relating the FoT to Mobile Ambients 85
3.13.3 Relating the FoT to Mobile UNITY 86
3.14 Expressing the FoT in π-calculus terms 86
3.14.1 Communications Network 87
3.14.2 Network Entity 87
3.14.3 Media 87
3.14.4 Communications Network Topology 88
3.14.5 Message exchange 88
3.14.6 Communication path 88
3.14.7 Location 89
3.14.8 Fixed network entity 89
3.14.9 Mobile entity 90
3.14.10 Mobile entity class 90
3.14.11 Network entity class 90

3.14.12 Complete entity class 91
3.15 Summary 91
4. Managing mobility in the Framework of Terms 93
4.1 Information viewpoint 94
4.1.1 Identity of mobile network entities 94
4.1.2 Unique identifier (UID) 96
4.1.3 Scope of uniqueness 98
4.2 Computational viewpoint 100
4.2.1 Mobile-Entity-Locator (MEL) Design Pattern 102
4.2.2 Mobile Entity Agent (MEA) Design Pattern 107
4.2.3 Combining the MEL and MEA design patterns 113
5. Mapping the Framework of Terms to real environments 116
5.1 A specification for Role Mobility in Prospect 117
5.1.1 The service environment in Prospect 117
5.1.2 Service roles and role mobility 119
5.1.3 Mapping the FoT to the Prospect environment 124
5.1.4 A specification of a role management component 126
5.1.5 Mapping the FoT requirements to the Prospect environment 135
5.2 An implementation of Role Mobility in Prospect 138
5.2.1 Mobility modelling in Prospect 138
5.2.2 Implementation use cases 139
5.2.3 Role Management sub-system 140
5.2.4 The Provider administrator Management User Application (PMUAP) 144
5.3 Mobile IP 145
5.3.1 Mapping the FoT to the Mobile IP architecture 145
5.3.2 Information viewpoint requirements 148
5.3.3 Computational viewpoint requirements 148
5.4 GSM 150
5.4.1 Mapping the FoT to the GSM architecture 150
5.4.2 Information viewpoint requirements 154
5.4.3 Computational viewpoint requirements 156
5.5 The TINA Architecture 157
6. Discussion 159
6.1 Definition of the FoT 160

6.2 Requirements on the FoT 162
6.3 Mapping the FoT to real systems 163
6.4 Mapping the FoT requirements to real systems 164
6.5 Drawbacks 165
7. Conclusions and further work 167
7.1 Conclusions 167
7.1.1 Experiences from the design of the role mobility component in Prospect 170
7.2 Further work 171
Glossary 173
References 179
Appendix A: CORBA design pattern template 199
Appendix B: Brief tour of the π-calculus notation 204

List of figures

Page

Figure 1-1: Schematic of the thesis approach....33
Figure 2-1: The operation of Mobile IP. 38
Figure 2-2: The operation of GSM – registration in a visiting country....45
Figure 2-3: The operation of GSM – call to a mobile station in a visiting country and originating from the visiting country....46
Figure 3-1: Three different communications networks as perceived in the same communications environment: (a) home user's view, (b) Internet Service Provider's view, (c) Telecom provider's view. 59
Figure 3-2: Different media between two hosts with an Ethernet physical connection..63
Figure 3-3: Example of CPs between network entities in different point-to-point topologies: (a) Star, (b) Ring, (c) Tree, (d) Complete, (e) Irregular....65
Figure 3-4: The location of an entity (B) when another entity (A) initiates a message exchange with it: (a) entity (B) is local to entity (A); (b) entity (B) is remote to entity (A). 68
Figure 3-5: Example of Fixed and Mobile network entities....71
Figure 3-6: Steps 1 … n of the analysis for fixed network entity identification requirements. 81
Figure 4-1: CP without network entities, where the sink entity is mobile. 94
Figure 4-2: CP with one network entity, where the sink entity is mobile. 95
Figure 4-3: CP with two network entities, where the sink entity is mobile. 96
Figure 4-4: CPs between network entities of different NECs and MECs. 98
Figure 4-5: A recommended combination of the MEL and the MEA design patterns. 113
Figure 5-1: Prospect Enterprise Model....118
Figure 5-2: Example of network entities and their locations including roles....122
Figure 5-3: States of a role-holder....124
Figure 5-4: One communications network in Prospect. 126

Figure 5-5: MECs and their corresponding CECs in a communications network of Prospect. 127
Figure 5-6: Role mobility and role management use cases. 129
Figure 5-7: Computational view of the role management system. 133
Figure 5-8: Exchange initiation between user and role in Prospect. 136
Figure 5-9: Role mobility and role management implementation use cases. 139
Figure 5-10: Component view of the role management system. 141
Figure 5-11: Main panel of the Provider Management UAP for Role Management. 143
Figure 5-12: Panel of the Provider Management UAP for setting-up roles. 144
Figure 5-13: Example of a Mobile IP communications network. 146
Figure 5-14: Example of MECs and corresponding CECs in a Mobile IP communications network. 147
Figure 5-15: Example of a GSM communications network. 152
Figure 5-16: Example of MECs and corresponding CECs in a GSM communications network. 153

List of tables

Page

Table 1-1: Mapping of terms used by different standards bodies. 22

Table 3-1: Identifiable FEs per message exchange per medium. 79

Table 4-1: Requirements for the identification of mobile network entities in the ANM. 100

Table 5-1: Information to support invitations and role-holder state management. 131

Table 5-2: Role description. 132

Table 7-1: Thesis contribution. 169

1. Introduction

The recent success of mobile telephony has given a clear indication that the users of communications services consider mobility a very important feature. Since then, there has been considerable effort from standards bodies to provide guidance on the deployment of mobile services. This effort comes from both the Internet and the Telecom communities and is discussed in the following section.

The starting point of the efforts of these communities has been the definition of the architecture of mobile systems and, at the same time, the interworking between these mobile systems and legacy systems. The Internet community deployed an architecture for the support of mobile IP nodes. Similarly, the Telecom community deployed architectures for the support of mobile telephony. The architectures of these mobile systems present many problems to be investigated and resolved.

First of all, most legacy communication systems were designed without consideration of mobility. The design of early communication services was based on the assumption that the parties involved in these services were fixed. For example, in traditional telephony it was assumed that the telephone terminals would be fixed in one building. Similar assumptions were made by the Internet community regarding Internet nodes. Trying to extend the architecture of these existing systems to support mobile communication parties often led to a trade-off between extensive re-design and continued support of legacy systems. As a result of this trade-off the mobile systems that emerged based on these legacies had certain problems. Many of these problems are common in the design of systems that come from completely different communities. For example, the problem of triangular routing is common in both the GSM architecture of the Telecom community and the Mobile IP architecture of the Internet community.

The initial assumption that the communication services would involve fixed parties is evident in different aspects of the design of the early communication systems. Since a communication party was assumed to be fixed to a particular location, the identifiers for these parties were also used to identify their location. The mechanisms for conveying communication messages to these parties relied heavily on the identification of their location and, as a consequence, to the identification of the parties themselves. Because of this, when a communication party changed location, they had to be given a new identity, which was the identity of their new location. For example, a home telephone number in traditional telephony is used to identify the person who is occupying the house were the telephone equipment is installed and also the telephone terminal and the end-point were the terminal is attached. The common practice of the Telecom companies has been that if a person moves to another home in another area they will have to be assigned a new telephone number, the number of the telephone of their new home.

Because of this coupling between the identity of a communications party and its location locating mobile parties was not supported efficiently. In our example of traditional telephony if a person moved home someone would have to obtain their new home telephone number manually, via the telephone directory, before being able to contact them.

However, considering mobility in the design of communication services requires more than just re-designing communication systems to fix existing problems with supporting mobile parties. It requires a different way of thinking about communication services and the parties involved in these services. It could be the case that if someone had to re-design the telephony service to accommodate mobile communication parties, they would repeat the same steps: design a system assuming that the parties were fixed first, and then extend this system so that mobile parties could be supported. Abandoning this kind of approach requires a *cultural* change which needs a wider discussion on the nature of mobile systems and on their differences to the fixed systems.

Eventually, this requires an exchange of ideas and a consensus on what mobility really is regardless of specific technologies and market background. In other words, a generic and technology independent definition of mobility is required.

Only at a generic level is it possible to investigate the essence of mobility. Also, only at a technology independent level is it possible for the different communities to discuss the problems of mobile system design. A generic and technology independent framework of terms for mobile systems could provide a common ground for the exchange of ideas.

Trying to define a framework of terms for mobile systems and investigating the requirements that mobility poses using the terminology of this framework leaves some open questions:

Q1. Can such a generic and technology independent framework of terms be defined?

Q2. Can this framework be generic enough to express a variety of systems and at the same time be of practical value?

Q3. Would any conclusions for mobility in this framework map to real systems?

Q4. Would such a framework assist to avoiding common pitfalls and problems of existing architectures in supporting mobility?

Our belief that the answer to the above questions can be positive inspired the hypothesis for this work.

In this thesis we provide a generic and technology independent framework of terms for mobile systems and we discuss the architectural aspects of mobile systems in this framework. Then, we investigate the hypothesis that this framework:

H1. Can express a variety of real mobile systems,

H2. Is of practical value, and

H3. Can provide for systems that avoid common problems of existing mobile system architectures.

In this chapter we look at the different approaches that different communities follow to identify mobility and to define mobile systems. We identify the problems that arise from the diverse definitions of mobility and mobile system architectures by the different communities. We then describe the hypothesis that motivated this thesis and the approach that was followed to investigate its validity.

1.1 Current standards and research

1.1.1 Internet

Starting with the Internet, a group of the IETF (Internet Engineering Task Force) has been working on the specification of the Mobile IP [RFC2002] protocol and the Mobile IP architecture, which provides for transparent routing of IP [RFC0791, RFC1883] datagrams to mobile nodes in the Internet. This work has influenced or encouraged other mobility-related activities and standards within the IETF [RFC2103, BPP96, PJ00, PJ98,Penn93, RFC2041]. Protocol extensions were suggested and protocol optimisations were proposed to cope with mobile nodes and Mobile IP [PJ00, PJ98].

1.1.2 Telecom

1.1.2.1 ETSI

On the Telecom side, the success of the Global System for Mobile communications (GSM) [GSM03.02] in delivering mobile telephony on a fully deployed commercial base in Europe and Japan has been one the most impressive achievements of the European Telecommunications Standards Institute (ETSI). The third-generation mobile systems as they are envisaged in Europe, will integrate the current second-generation systems and provide a new range of broadband services, which will also be compatible with the bearer and advanced services of the fixed telecommunication networks [DF95].

In order to do that, ETSI recognises that very soon GSM will have to be extended or integrated into a more generic framework that will allow for a wider variety of services. This requires more bandwidth and a network that will have to be better integrated and managed with the fixed core network and satellite systems [UMTSsat, UMTSmgmt, DF95, Rapel95]. The proposed standard is called Universal Mobile Telecommunications System (UMTS) and ETSI expects that it will be developed in some form during the first decade of the 21st century. There is ongoing work on the standardisation of the UMTS architecture [KPMF98, FHVN98].

1.1.2.2 ITU

Apart from ETSI, the International Telecommunication Union (ITU) has been working on a new family of standards under the name Future Public Land Mobile Telecommunication

System (FPLMTS) [M.687-1], which was recently renamed to International Mobile Telecommunications 2000 (IMT-2000) [M.687-2]. These standards are the work of the ITU Radiocommunications Standardization Sector (ITU-R) and of the ITU Telecommunications Standardization Sector (ITU-T). ITU expects to complete this family of standards around the year 2000. IMT-2000 envisages an environment where users will be able to access a wide range of voice or data services anywhere, anytime by means of a small pocket communicator. Apart from this, IMT-2000 tries to unify existing diverse systems (fixed or mobile) so that the user perceives a Virtual Home Environment (VHE) where the same services can be received regardless of the media or terminals used [BFMC97]. ETSI and ITU have been exchanging ideas about the future deployment of third-generation mobile systems. This exchange is apparent in the work of the Special Mobile Groups of ETSI, where ideas of IMT-2000 on numbering and the VHE are discussed. Also, ETSI is considering the use of the Telecommunications Management Network (TMN) ITU-T standard [M.3010] for the management of mobile services [UMTSmgmt].

An earlier standard from ITU-T on Universal Personal Telephony (UPT) [F.850, E.168, I.114] has provided guidelines on how users can access services regardless of the terminal used by means of a unique identifier that is associated with each user and which is used to make and receive calls. UPT was initially intended to provide personal mobility on the fixed telephone network. It was part of the ITU standardisation work for the support of UPT in the Intelligent Network Architecture (IN) [I.373, I.312, I.318, Q.1290]. The IN architecture has already achieved the provision of additional capabilities in telecommunication services independent of service, network implementation, or vendor equipment. Examples are the multi-party conference and the freephone services. IN achieves this by introducing a separation between service logic and the network.

1.1.2.3 The TINA Consortium

Influenced by the IN and the UPT standards, the Telecommunications Information Networking Architecture (TINA) consortium [DNI95] has worked for the past five years on an object-oriented Service Architecture [TINA-SA5, TINA-SAA5] that provides a set of concepts, principles and guidelines for the deployment and operation of advanced communications services. The TINA consortium (TINA-C) tried to build on previous standardisation efforts such as the Telecommunications Management Network (TMN) and the Intelligent Network architecture of ITU. TINA attempts to bring into the Telecom community the ideas and concepts of the Open Distributed Processing Reference Model (RM-ODP) [ISO-ODP1, ISO-ODP2]. For this reason, TINA suggests an integration of telecommunications systems with the recent advances of distributed computing such as the Common Object Request Broker Architecture (CORBA) standards by the Object Management Group (OMG) [CORVAv1, CORBAv2, CORBAsvc, Kitson95]. The TINA proposals advocate object oriented design and reusable software components for the rapid deployment and management of "open" services.

The TINA Service Architecture (SA) is part of an overall architecture [TINA-overv], which clearly separates services from communication networks, so that communications services can be introduced and deployed regardless of the underlying communications network technologies. These separations come from a requirement analysis that the TINA consortium has produced [TINA-req]. The TINA SA provides for personal and session mobility support. The former provides for users being allowed to use the TINA services regardless of the terminal that they are attached to. The latter provides for users to be able to suspend their service sessions and later resume them from another terminal. From its beginning, TINA addressed the management requirements for mobility. The scope however, has been how the next generation services can be managed in the TINA framework [RF95].

Many European projects of the ACTS (Advanced Communications Technologies & Services) European research framework such as VITAL and Prospect have realised TINA or TINA-like systems. In VITAL TINA has been extended with a Discrete Terminal Mobility (DTM) service which allows for terminals to be detached from one network access point and attached to another, while maintaining the facilities that they offer [VITAL-D13].

Earlier, European projects like PERCOM investigated personal mobility, session mobility and what they define as "user environment mobility" in the framework of a Personal Service Communication Space (PSCS) of the European RACE programme (Research for Advanced Communications in Europe). These projects imported principles similar to the TINA ones from early on. Their approach was to provide and demonstrate mobility support over a number of different technologies, such as PSTN, NB-ISDN, GSM, BB-

ISDN [HC95]. Many of these projects contributed towards evolving the IN architecture to the TINA architecture but their approach applies mostly to Telecom environments.

The TINA work on mobility is now continuing a special TINA Next Generation Mobility Work Group (TINA-ngm-wg).

1.1.3 Other research

1.1.3.1 ODP

The series of standards for Open Distributed Processing (ODP) by the International Organisation for Standardisation (ISO) and ITU-T provides a meta-model for *open* distributed systems [ISO-ODP1, ISO-ODP2, ISO-ODP3, ISO-ODP4].

This model suggests an object-oriented approach for deploying distributed systems. The term *object-orientated* refers to objects that encapsulate functionality in their implementation, and that, through *interfaces*, offer an abstract view of this functionality in a system. In their book on "Open Distributed Processing and Multimedia", Gordon Blair and Jean-Bernard Stefani give a definition of a distributed system as one that is: "*designed to support the development of applications and services which can exploit a physical architecture consisting of multiple, autonomous processing elements that do not share primary memory but cooperate by sending asynchronous messages over a communications network*" [BS98]. The attribute *open* in a distributed system implies certain rules to ensure interoperability and portability by having the objects of this system conform to well-defined interfaces.

The ODP architecture identifies five viewpoints from which a distributed system can be described or specified and a number of transparencies that open distributed systems can support. Among those, there are *location transparency*, *migration transparency* and *relocation transparency*. Location transparency hides the physical location of an object by means of a unique identifier. Migration transparency allows an object to move to another location transparently to its client, while relocation transparency allows the object to migrate even while it is involved in an interaction with other objects.

ODP is a *meta-model* [BS98] that does not elaborate on how these transparencies will be implemented; it only intends to provide a framework for specifying open distributed systems. Therefore, apart from the suggestion of unique identifiers, no other aspects for mobility are presented or justified.

1.1.3.2 Mobile Agents

The term mobile agent has been used extensively by the Internet community, by the distributed computing community and by the Telecom community [PK98]. Although there is no clear definition of a mobile agent, it is certain that there are two aspects to it: agency and mobility [VT97].

Starting with the *agency* aspect, it is implied that a mobile agent as a software entity can act on behalf of a user or another entity. In other words, a mobile agent can *delegate* for a user or another entity. It may be an intelligent agent [PK98] or it may have to do mundane tasks on behalf of the other entity or user that delegates to it. Agents have been suggested for network management [Yemini91], for delivering services [KM96, BM98], or for service engineering [GD98]. Agents have also been deployed to offer personalisation to users and personal mobility over wireless or wireline networks [INM95]. From the mobility perspective though, these approaches focused exclusively on personal mobility and they were not intended to address mobility in a wider scope.

The aspect of *agent mobility* refers to allowing an agent to migrate, on its own, closer to the source of the data that it has to process. There have been several proposals for how this would benefit network management or service delivery [PK98]. However, the main objective of all the approaches to agent mobility is to reduce the network traffic, increase fault resilience, or allow local access of data, by moving the agent closer to the source of the data. This work relates to the research for process migration [ANSA, CORBAsvc] but it focuses to solving these specific problems.

Mobile agents that can delegate for service users and provide personalisation have been proposed in the past [RLV95]. Again, the objectives of these proposals were to move the personal mobility agents closer to the source of the data so that the network traffic is reduced. This approach is considered particularly efficient for traffic that otherwise would be transmitted from a low power capacity mobile terminal over a low bandwidth wireless link [PK98]; other approaches to reduce this kind of traffic involve use of proxies or caching. All, these proposed approaches mainly intend to resolve the particular technology problem of data traffic over a low power capacity mobile terminal.

1.1.3.3 European research on mobility

The two recent European research frameworks ACTS and RACE have endorsed the concepts that ETSI and ITU are jointly working on and they have commissioned several projects with the objective to contribute to this work. In the ACTS framework a user-trial driven approach was used in order to assess advanced mobile services. Many of the ACTS and RACE projects investigated how technological problems can be overcome in order to deploy these advanced services as envisaged in UMTS and FPLMTS [DIE97, FHVN98, DF95].

The RACE project MOBILISE defined the concept of the Personal Service Communications Space (PSCS) that was reused in other projects of the RACE and of the later ACTS European frameworks [GKGF95]. The PSCS architecture is based on the Intelligent Network (IN) architecture and it aims to provide personal mobility and personalisation on a variety of communication services over different technologies.

Other European projects like the ACTS project Multicube provided solutions to how UPT concepts of personal mobility and personalisation can be supported in Internet-based multimedia conferencing services [Schul96].

Projects of the ACTS framework that investigate personal or terminal mobility have collaboratively produced a number of guidelines for mobile systems. These guidelines were produced under the NIF (Fixed and Mobile Interworking) chain activity of the NI (Network Interworking) chain group of ACTS. The chain groups of ACTS offer a forum for the exchange of ideas across ACTS projects in certain areas. The NIF chain guideline number 3 (NIF-G3) [NIF-G3] addresses personal and terminal mobility requirements in a TINA/IN like Service Architecture under the name Common Service Architecture (CSA). A service engineering approach to the deployment of mobile services is discussed in this guideline.

1.2 The problem area

From this brief overview of the standardisation efforts by various communities it becomes evident that different paths are being followed for the specification and provision of mobile services. A study of the existing standards makes it clear that there is lack of communication between the different bodies, which are using different languages to address mobility, the problems that mobility implies and the solutions for these problems.

Table 1-1 attempts a mapping among different terms used by different standards communities in relation to mobility.

Internet (Mobile IP)	ETSI (GSM)	ITU (UPT)	ITU (IMT-2000)	RM-ODP	TINA
user	subscriber	user	user	n/a	user
node	station	terminal	station	node	terminal
router	location area	n/a	n/a	n/a	network access point
n/a	IMSI	UPT number	IMUI	identifier	user ID
mobile node IP-address	MSISDN	n/a	IMSISDN	identifier	terminal ID
mobile node care-of-addr.	MSRN	n/a	IMSRN	n/a	network address
MAC address	IMEI	n/a	IMEI	n/a	n/a

Table 1-1: Mapping of terms used by different standards bodies.[1] [2]

***Problem 1:** The different standards communities define mobility differently and they employ different language/terminology when referring to mobility.*

The Internet community addresses only the problem of IP node mobility. The Mobile IP protocol aims to provide for the mobility of IP nodes in an IP network. Currently, there are no recommendations to address other mobility areas. For example, the problem of personal mobility and how to allow Internet users to use different Internet terminals without problems is not addressed by the Internet community.

On the other hand, the standards of the Telecom community perceive mobility differently. The ETSI GSM recommendation provides partly for personal mobility by allowing a user to change terminal by means of a Subscriber Identity Module (SIM) card [GSM02.17]. It provides for terminal mobility by prescribing the hand-off and signalling procedures when a mobile terminal moves from one cell to another or from one mobile network to

---

[1] Some of the mappings among UPT, GSM and IMT-2000 were taken from the paper "IMT-2000 Standards: Network Aspects" [PGLM97].

[2] The mapping of a GSM IMEI to an IMT-2000 IMEI is based on the assumption that the GSM IMEIs will be supported in the framework of IMT-2000. However, ITU has no intention at the moment to produce an international standard for IMEIs [PGLM97].

another. This approach is confined within the domain of the mobile telephony since there is no provision for the users or the mobile terminals to flexibly attach to the fixed network.

The communities that are active in the mobile agent field examine the problem of mobility only in the narrow scope of process migration. Mobility is not addressed in a generic way. Mobile agent platforms and technologies are mainly envisaged as an infrastructure that will extend the client-server paradigm [PK98, RLV95] in various applications.

> ***Problem 2:*** *Each community addresses only some kinds of mobility and they follow a parochial approach to the problem.*

The differences between the proposed architectures can be attributed to the legacies of each side and the bottom up approach that has been followed. Both ETSI and ITU state that a top-down approach is being followed for the specification of UMTS and IMT-2000, but they have not provided concrete output yet. Also, the approaches that were followed in both the specification of Mobile IP and the GSM have certain problems, which are described in the following chapter.

> ***Problem 3:*** *Many of the current mobile system architectures have problems.*

There is a striking difference between the Telecom and the Internet communities in the scope over which they address each kind of mobility. For example, in the GSM standard it is implied that the problem of terminal mobility is addressed by allowing mobile terminals to move from one cell to another or from one area to another. However, terminal mobility for wireline terminals is not addressed. Also, in the GSM standards the identity that is used in order to locate a terminal is an International Mobile Station Identifier (IMSI). But in the same standards, the IMSI is also associated with a user. Therefore, it is not clear whether GSM allows for terminal mobility or for mobility of a "subscriber – terminal" tuple.

Similarly, in the Mobile IP specification an IP address is used to identify a node uniquely. This seamlessly allows a Mobile IP node to change its point of attachment to the Internet. But on the other hand, the examples of IP usage in the specification are about how to allow a user to move with a laptop terminal to a different point of attachment to the Internet [RFC2002, PJ00]. It is not clear again whether this is for terminal mobility or for personal-terminal mobility. What is certain, is that there is no Internet standard to allow a user to move to a different terminal on another network without having to set up a new environment. That could be seen as a basic requirement for personal mobility.

Some of the concepts from the mobile agents research community, such as unique identification for mobile agents or security issues regarding agent execution in visiting environments, could address generic aspects of mobility. However, the objectives of the mobile agent work, which are mostly concerned with reducing network traffic [PK98], certainly place a limit to the scope in which these mobility-related areas are addressed.

> ***Problem 4:*** *The current mobile system architectures are not only parochial in the kinds of mobility that they address, but also in the scope in which they address each of these kinds of mobility.*

For these reasons, we consider that a generic and technology independent definition of mobility is missing today. Standards bodies use different language to define mobility and they have provided partial solutions and of a limited scope. We believe that a common language and a common perception of mobility are essential.

> *A generic and technology independent framework for mobile systems could provide the infrastructure for addressing mobility in its whole and for exchanging ideas and experiences on solving mobility problems regardless of technologies and architectural legacies.*

A generic and technology independent framework of terms could provide solutions for the problems that are mentioned above in the following way:

S1. With regard to the different terminology that is employed by different communities when addressing mobility (problem 1), a framework of terms could provide a basis on which communities can agree on a definition of mobility and it could provide a common language and terminology that allows them to discuss mobility in common, generic terms.

S2. Also, by addressing mobility in generic terms the different aspects and kinds of mobility can be defined in a generic way. This way, parochial definitions of the kinds of mobility that individual communities make (problem 2) can be avoided.

S3. As far as the architectural problems of existing mobile systems are concerned (problem 3), a common framework could provide the basis on which solutions to common mobility problems can be expressed and exchanged between different communities in a generic form. This could assist in building coherent systems that avoid the common pitfalls of existing mobile system architectures.

S4. Finally, defining different kinds of mobility in terms of a common framework will make it easier for communities to identify the scope in which they address each kind of mobility (problem 4).

The approach of a common terminology framework for mobility could provide a common point of reference when addressing mobility issues. Experiences and approaches could be expressed and understood between different cultures, such as the Internet and the Telecom.

A generic and technology independent framework would have a longer lifecycle than technology dependent ones and, therefore, it is expected that it would serve its purpose in the long term. In this way, we could hope that customers would receive better services and that the mobile service providers would be able to be more competitive.

There are also advantages in keeping the definitions of this framework *informal* rather than providing formal definitions. First of all, formal definitions could be of limited practicality. Not every mobile systems designer is familiar with formal methods and not everybody understands a formal specification. By providing an informal terminology it is possible to communicate ideas to a wider audience and also to receive feedback from an audience that may have valuable experience in mobile systems design. On the other hand, informal definitions have the problem that they can be ambiguous and that they cannot provide ground for rigorous theoretical work.

The idea of providing a framework of terms for mobile systems on which we can outline solutions to building efficient mobile systems inspired our hypothesis and initiated the work towards investigating its validity. In particular, a number of questions arose regarding the use of a framework of terms.

Questions:

Q1. *Would a framework of terms for mobile systems offer a generic definition of mobility? Would it provide a common language and terminology for mobility?*

Q2. *Is it possible to construct a generic and technology independent framework that is sufficiently expressive to be capable of describing/representing a generalised model of mobility but not so loose that it is of no practical value?*

Q3. *Would any conclusions for mobility on this framework map to real systems?*

Q4. *Would such a framework assist to avoiding common pitfalls and problems of existing architectures in supporting mobility?*

By believing that the answer to the above questions can be positive, we expressed a hypothesis and we planned the work to be done towards validating it. The hypothesis is presented in the following section.

1.3 Hypothesis

Starting from the realisation that the existing standards fail to provide generic solutions to the problem of mobility, it was decided to investigate if mobility issues can be described and solved in a generic and technology independent way. This was needed to gain an insight into the essence of mobility. In order to address mobility in this way, mobility had to be studied at an abstract level. The hypothesis that motivated this work was that:

A framework of terms for mobile systems can be deployed, which:

H1. *Is capable of providing a generalised model for mobility, on which mobility can be defined and requirements for it can be investigated.*

H2. *Although generic and technology independent, it is of practical value. It can sufficiently express real systems and it can show where a real system may fail in supporting mobility, and why.*

H3. *Can provide for building coherent mobile systems that avoid common pitfalls of existing mobile system architectures.*

Investigating the validity of this hypothesis would open the way for the exchange of ideas on mobile systems at a generic level, with the anticipation that these ideas would also apply to the real systems to which the terms of the framework can map.

The approach that was planned and followed to investigate the validity of this hypothesis is described in the following section.

1.4 Thesis structure and approach

In this section the structure of this thesis and the approach that was followed in investigating the validity of the hypothesis are discussed. A short introduction to the concepts and tools that were employed for this investigation is also provided.

1.4.1 Overall approach

In this thesis we will describe generic Framework of Terms (FoT) for mobile systems which is technology independent. Within this framework we consider a number of mobile entities and we investigate the requirements for allowing mobile entities to contact or be contacted by other network entities, fixed or mobile. Instead of describing mobility in terms of "fixed internet host", "mobile terminal", "wireless link" or "protocol" we discuss mobility in terms of "network entity", "mobile network entity", "medium" or "message exchange". It is within this generic framework that we outline the concept of mobility.

This allows us to investigate mobility at a high level and to have a broad view of the problem. The intention at this level is to discuss the minimum requirements for mobility and to induce a number of requirements that must be considered for the deployment of mobile systems. It is not intended to provide prescriptive guidelines or a methodology for deploying mobile systems. The intention has been to provide a kind of checklist for potential problems in systems that are to provide mobility.

Apart from providing a framework for mobile systems, another goal of this thesis was to test the generality and practical value of such a framework. A first test was to map this framework of terms to a number of real mobile systems from different communities in order to investigate its generality. A second test was to investigate the practical value of this framework by investigating if a real system can be deployed based on this framework and respecting its requirements and if the real system could support mobility efficiently. Additional to this, it was tested if well-known problems of existing mobile systems can be justified by looking at them through this framework. This would further enhance the practical value of our framework and also demonstrate that it can provide for coherent systems that avoid common pitfalls of existing architectures. In order to reduce the ambiguity of some definitions in this framework, which is due to its informal nature, some of its concepts are also expressed using a formal notation.

1.4.2 Concepts and tools

The OSI reference model, the ODP viewpoints and UML were the essential tools and the background for the approach that was followed. A short description of these and how they were employed in this thesis is given in this section.

1.4.2.1 OSI Reference Model

The Open Systems Interconnection (OSI) Reference Model (OSI-RM) [OSI-RM] of the International Organisation for Standardisation (ISO) and ITU-T addresses interconnection issues between heterogeneous systems. The model is structured in seven layers that provide different levels of abstraction for a network connection. Since the scope of the OSI-RM is limited to inter-connection issues between different systems [FLM95], the OSI-RM is not necessarily related to mobility. However, it is an example of a generic model that has succeeded as a platform for the exchange of ideas and specification of system inter-connection both in the Internet and the Telecom communities.

The Framework of Terms (FoT) of this thesis is orthogonal to the OSI-RM. The FoT intends to subsume mobility in an abstract communication environment, where the connections between entities are taken for granted and they are not described as in the OSI-RM.

1.4.2.2 ODP viewpoints

The RM-ODP is an ongoing joint standardisation activity by ISO and ITU-T which provides a technical framework to describe and standardise the common features of systems for Open Distributed Processing [ISO-ODP1, ISO-ODP2, ISO-ODP3, ISO-ODP4]. The RM-ODP introduces the concept of a "viewpoint" to categorise the various concerns of distributed systems. Five viewpoints are specified: enterprise, information, computational, engineering and technology. The enterprise viewpoint relates to the enterprise entities in a distributed system and the business roles and responsibilities between them. The information viewpoint focuses on the structure and the transformation of information in a distributed system. The computational viewpoint addresses the logical partitioning of distributed applications into components and the interfaces between them. The engineering viewpoint gives a system oriented view of distributed applications and distribution transparencies, while the technology viewpoint is concerned with the technologies that implement the engineering mechanisms [FLM95]. It is considered that the five viewpoints of ODP can capture any concern of a distributed system [FLM95]. The RM-ODP is intended to provide an application oriented reference model, while the OSI-RM has a scope that is limited to the communication protocols of networks.

From the definition of distributed systems of section 1.1.3.1 it can be concluded that *a mobile system is by its nature distributed*, since the mobile entities of any system will have to be autonomous processing elements that cooperate via messages with other entities. Should the "mobile entities" share resources with the other entities that they communicate with, mobility is not an issue.

It is considered that the requirements for mobility that are investigated in the FoT *can be described* and understood if they are examined from the ODP viewpoints.

However, the use of ODP viewpoints in this approach to a mobile system *does not imply* that a mobile system expressed in the proposed framework of terms *has to be an open distributed system*, and *it is not required* or attempted to investigate *ODP conformance* in the FoT. Also, it *was not* considered necessary to *adhere to object-orientation principles* as defined in programming languages like C++ and Java or in standards like ODP or CORBA.

1.4.2.3 The use of UML

The Unified Modelling Language (UML) serves as a tool for object-oriented modelling for different kinds of systems [UMLv1s97, UMLv1n97]. It unifies similar modelling languages and techniques, like the Object Modelling Technique (OMT) [Rumb91] or the Booch method [Booch94]. It is also intended to cover all the aspects and stages of systems development, from the requirements analysis to the final implementation. In this thesis UML notation was used to describe a mobile system. However, the methodology of the design did not strictly follow the UML proposals. Here, UML is only used as a tool for describing systems. Methodologies for mobile system deployment are out of the scope of this work.

1.4.2.4 Design Patterns

It can be observed that certain problems recur during a design process, like in the design of a building or in the design of a system, even in the writing of a novel. Gamma et al [GHJV97] note that, during the design of an object oriented system, certain problems keep reappearing. In order to solve these problems, the system designers reuse solutions that they employed for similar problems in the past. Since these solutions come from past experience, they have been tested, and their pros and cons are known to the system designers. These reusable solutions constitute the experience of a systems designer.

Gamma et al point out that if these problems and their solutions could be described in a generic way, then it would be easier for systems designers to exchange experiences on common design problems [GHJV97]. Also, they could create a knowledge base of common solutions for these common problems. What is suggested is a template for describing the problems and their solutions. These "solutions" are called design patterns. A design pattern template allows for identifying a pattern, classifying it, describing the problem it solves, describing the solution, and presenting the pros and cons of that solution. Gamma et al started a catalogue of design patterns by providing a set of approaches for common object-oriented software development problems [GHJV97].

Later on, Mowbray and Malveau point out that scale is very important in design patterns [MM97]. Design patterns could be used to address software architectures at different levels, from a *microarchitecture level* that concerns patterns for fixing small problems in software applications up to a *global level* that addresses software that crosses the boundaries between enterprises. Mowbray and Malveau specify a special design pattern template that accommodates this concept of scale, and they present a number of design patterns, each of which can apply to one or more levels of this scale. Their work presents design patterns for applications of the Common Object Request Broker Architecture (CORBA) by the Object Management Group (OMG) or Java applications [GJS96].

Although the design pattern template by Mowbray and Malveau is intended for distributed systems software, it was found suitable to describe design patterns that involve entities of the framework of terms of this thesis. This does not imply that the entities of the FoT are necessarily distributed software entities. Also, although it does not imply any association with CORBA or Java, these could be natural implementations of instances of the design.

Another difference in the design patterns of this thesis is that they describe requirements for mobility rather than experience with mobility problems. They are the result of a top-down approach to the problem of mobility at an abstract level, rather than the result of bottom-up experience with mobile system deployment. This type of design patterns are often called *Analysis Patterns*. Part of this thesis work is to apply them to real systems, and to see how these systems then perform.

1.4.3 Detailed approach

The approach that was followed to investigate the validity of the hypothesis that motivated this thesis consists of four steps as illustrated in Figure 1-1.

S1. The first step was to specify a Framework of Terms (FoT). This framework was designed to be generic enough to encompass concepts found in a variety of existing mobile systems. The definitions of this framework were also intended to be technology independent. This framework was intended to provide the base for a discussion on mobility and not to abstract the concept of a network connection like the OSI-RM. It describes the bare characteristics of a communications environment that features mobile entities. The requirements for mobility support that are expressed in this framework of terms is only concerned with how mobile entities can contact or be contacted by other fixed or mobile network entities, where the connectivity and the data-transport between them are assumed. The term *entity* is generic enough to include network equipment (such as terminal equipment), persons, or applications. Due to its informal nature the definitions of framework can be ambiguous. In order to reduce the ambiguity in the framework its definitions are also expressed using a formal notation.

S2. The second step of the approach was to discuss mobility in the FoT and to see what are the minimum requirements for mobile entities. This discussion is based on the Open Distributed Processing (ODP) viewpoints of the RM-ODP.

In this thesis, requirements for mobility in the proposed FoT are investigated from two different ODP viewpoints mainly: the information viewpoint and the computational viewpoint. The enterprise viewpoint is not addressed; the main reason for this is that potential requirements regarding the enterprise configuration of a mobile system expressed in the FoT would take a lot of effort to test. It would take a thorough investigation of a variety of existing enterprise configurations. For the scope of this thesis, the FoT was intended to be generic enough so that it can map to single-enterprise mobile systems as well as to mobile systems with complex enterprise models. Only a few issues from the engineering viewpoint are discussed together with the computational issues. These issues are only discussed as different distribution options for the computational entities of the proposed design patterns and the potential advantages or disadvantages that these options may hold. The technology viewpoint is not included in the FoT since the FoT is technology independent.

Within the information viewpoint the use of identifiers for mobile entities and the necessity for unique identifiers are discussed. Also, a high-level description of the format they need to have is proposed. On the computational viewpoint the minimum number computational entities that are required in order to support mobility efficiently are examined. These entities and their minimum functionality are presented in the form of Design Pattern templates. In particular, the CORBA design pattern template of Mowbray and Malveau [MM97] is used, since that one seemed most adequate for expressing design patterns of a very wide scope and applicability. Using these design patterns some additional issues of an engineering level nature are briefly discussed. These are the different distribution options that can be used for some computational entities and the pros and cons of each option.

S3. The third step of the approach was to examine if the requirements for mobility support in the FoT could apply to real systems too. To accomplish this, the FoT was mapped to the mobile system that was being deployed in the European ACTS project Prospect [LTMR97]. Prospect deployed a system that offers personal mobility [TMK99] and role mobility [Tirop98, TMK99] in a multi-service TINA-like environment. The specification for the implementation in Prospect was carried out using the UML specification tools. Although UML diagrams are employed to present the specification of Prospect in this thesis, the methodology that Prospect followed to respect the FoT requirements is out of the scope of this work. This is because the FoT and its requirements are not intended to be prescriptive but descriptive of a mobile system. Therefore, it is only shown *where* the requirements were respected in the Prospect specification and not *how* the Prospect methodology applied them to the specified system. Additional to the Prospect work, a survey of well-known problems of existing mobile systems such as the GSM and the Mobile IP architectures was carried out. The FoT was mapped to these systems in order to examine which of the requirements of the FoT these systems fail to satisfy. Also, it was investigated if the FoT and its requirements could provide an insight to how some of the well-known problems of these architectures could have been avoided.

S4. The fourth step of the approach was to see if the Prospect system would be an efficient mobile system after considering and respecting the FoT requirements in its design. We present how role mobility was deployed and demonstrated in Prospect.

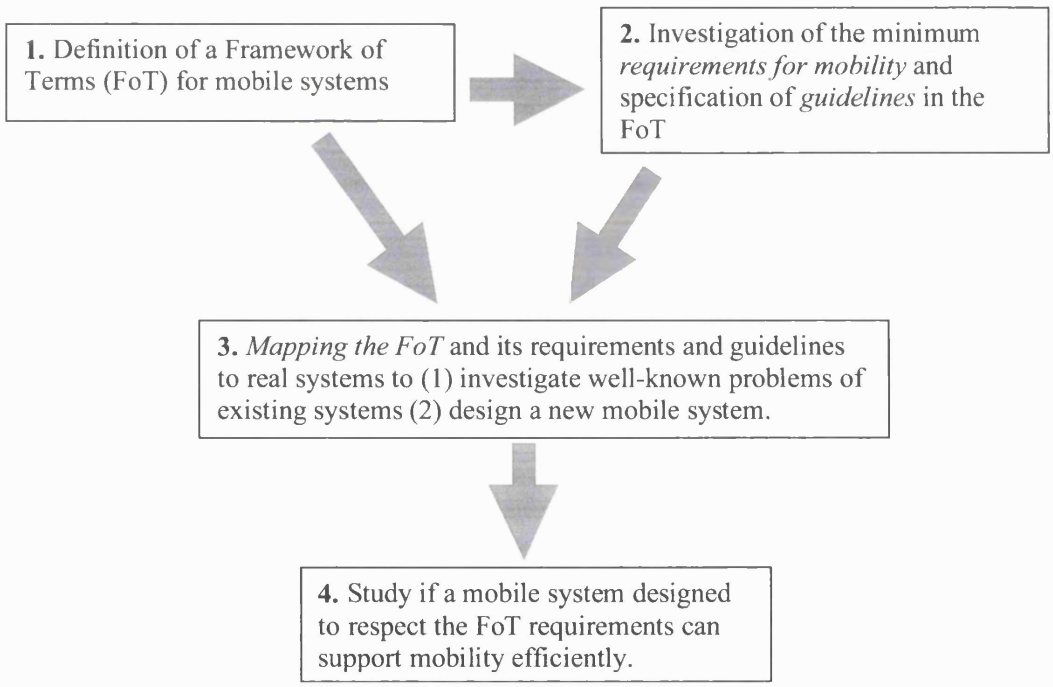


Figure 1-1: Schematic of the thesis approach.

This approach was able to test all the points of the hypothesis. The definition of a framework of terms for mobile systems and the definition of mobility in this framework is part of Step 1 and Step 2. The generality of this framework is tested by mapping it to a number of mobile systems coming from different communities (Step 3). The practical value of this framework is investigated by designing and demonstrating a real mobile system according to the requirements of the FoT; this system is the role mobility component in Prospect (Step 3 and Step 4). Also, the practical value of the FoT was further investigated by justifying well-known problems of existing mobile systems by showing where these systems fail the requirements of the FoT; the problems of Mobile IP and GSM are investigated in this way (Step 3). By showing that well-known problems of existing mobile system architectures can be justified by their failing certain requirements of the FoT, the claim that the FoT and its requirements provide for building coherent mobile systems that avoid pitfalls that are common in existing ones was substantiated.

Having demonstrated and investigated the validity of the hypothesis some conclusions are drawn regarding the FoT and the specified requirements and further directions are identified to carry this research forward.

The following section provides an outline of this thesis that documents this work in detail.

1.5 Outline of dissertation

In Chapter two and Chapter three we examine the problems with existing mobile system architectures and we present a generic and technology independent Framework of Terms (FoT), which is the first step of the thesis approach:

- In **Chapter two** we examine the architectures of GSM and Mobile IP and we point out the flaws and the problems that they present. We show why the scope of mobility is limited in these architectures. We point out the restricting role of legacy technologies in the existing standards and we also discuss how this legacy is passed on to the most recent standardisation work of the Telecom community. In the same chapter we also present other work that is relevant to this thesis, including existing abstract network models and mathematical models for mobility.
- In **Chapter three** we present the Framework of Terms (FoT) that defines mobile entities in a communications environment. We define the concept of a mobile entity, the media for communication exchanges between entities and the requirements that mobility introduces in this environment. Throughout this chapter, we show how the proposed framework of terms can map to existing mobile systems with a variety of examples. In chapter three we also relate the FoT to existing abstract models and

mathematical models for mobility. Since the FoT provides informal definitions that are expressed in plain English it can be ambiguous. In order to reduce this ambiguity we express FoT definitions using a formal notation at the end of this chapter.

- Then, we proceed with the documentation of the second step of the approach, in **Chapter four**. In this chapter we present a set of requirements for mobility based on the FoT. We present and discuss these requirements from two different ODP viewpoints: the information viewpoint and the computational viewpoint. Based on these requirements we proceed with defining architectural requirements for mobility at this generic level. From the information viewpoint we investigate the necessity and the scope of unique identifiers for mobile entities and we give a high level view of the structure that mobile entity identifiers should have. From the computational viewpoint we outline different kinds of computational entities that can assist in managing mobility efficiently. These are described by means of design pattern templates. Together with the computational viewpoint considerations we discuss engineering viewpoint issues.

We then go on to show how the requirements of Chapter four were met in the design of a real system and we present how this system was implemented and demonstrated. These are the last two steps of the followed approach and they are presented in Chapter five:

- In **Chapter five** we describe the deployment of a role mobility system that was designed and implemented under the European ACTS project Prospect. We show how the FoT of Chapter three maps to the Prospect environment and we present how the FoT requirements of Chapter four were considered in the design of this system. Finally, we present how role mobility was implemented and demonstrated in Prospect. Apart from the Prospect architecture, we show, in the same chapter, how the FoT maps to the GSM and IP architectures that were discussed in Chapter two. We show which of the FoT requirements these architectures fail to respect, and we explain how respecting these requirements in their design could cure the well-known problems of these architectures.

We discuss the results of this investigation with respect to the hypothesis in **Chapter six**, where we also investigate the contribution and the limitations of this work. The conclusions are presented in **Chapter seven** together with other aspects of this work that can be addressed in the future.

2. Current mobile system architectures and related work

In this chapter we describe two current mobile system architectures that originate from different communities:

A1. The Mobile IP architecture of the Internet.
A2. The GSM architecture of ETSI.

These two architectures are considered to be representative of each of the two cultures from the Internet and the Telecom communities. Both of these architectures evolved from legacy architectures. In both cases, mobility support had to be added to existing fixed network architectures.

For each of these architectures we present their problems and we discuss how their legacy led to their respective problems with regard to mobility support. Then, we extend our criticism to other architectures that are currently considered by the Telecom community. We focus this criticism on the following points:

P1. They address only some kinds of mobility.
P2. They approach each kind of mobility in a parochial way.
P3. Their approaches cannot be easily adopted for mobility in other environments.
P4. They have certain architectural drawbacks.

In this chapter we also discuss other approaches to architectural requirements for communication systems, which relate to this thesis. We present how the ODP viewpoints have been used for identifying requirements for multi-media systems, the relevance of our work to the OSI reference model and other work on the use of unique identifiers for addressing and locating network entities.

Finally, we present existing formal and mathematical models for mobility. In particular we present the π-calculus, the mobile ambients calculus and the Mobile UNITY.

2.1 Mobile IP

The Mobile IP protocol and its architecture [RFC2002, RFC2005] allow for transparent routing of IP datagrams to mobile nodes in the Internet. The architecture introduces the following new entities to IP networks for mobility support:

E1. A *mobile node*, which is an IP host or router that changes point of attachment from one IP network or subnetwork to another but can still communicate with other Internet nodes. The mobile agent maintains its IP address at all times. The IP address is used to identify the mobile node [RFC2002].

E2. A *Home Agent* (HA), which is a router on the home network of a mobile node, which, when the mobile node is not in its home network, "tunnels" the IP datagrams to it at its remote location.

E3. A *Foreign Agent* (FA), which is a router on a network where a mobile node can visit and it provides routing services to the mobile node while visiting. In this way, the mobile node can send datagrams to other Internet nodes via the FA. The FA can also "detunnel" all the traffic destined to the mobile node (this traffic arrives via the HA), and forward it to it to the mobile node.

Both the Home Agent and the Foreign Agent can be described with the term *Mobility Agents*.

When a mobile node moves from its home network to a foreign network the following steps take place:

S1. The mobile node realises that it is in a foreign network either by receiving the FA advertisements or by soliciting an advertisement from the FAs.

S2. The mobile node then obtains a care-of-address from the foreign network that can be used to receive IP datagrams.

S3. The mobile node registers its care-of-address with its HA by sending a registration message to it, possibly via the FA.

S4. After this, datagrams destined to the mobile node are intercepted by its HA and subsequently tunnelled through to the care-of-address. If the latter is a FA address, the FA will detunnel the datagrams and forward them to the mobile node. If the care-of-address is bound to a network interface on the mobile node itself, then the mobile node will do the detunnelling. Outgoing datagrams from the mobile node are routed in the standard way via the foreign network.

S5. Every time that the mobile node moves to another foreign network or to its home network it has to update its registration with its HA.

When at home, the mobile node operates without mobility services.

Figure 2-1 illustrates steps 1 to 4. This diagram illustrates most of the architectural entities that Mobile IP introduces and it outlines their operation.

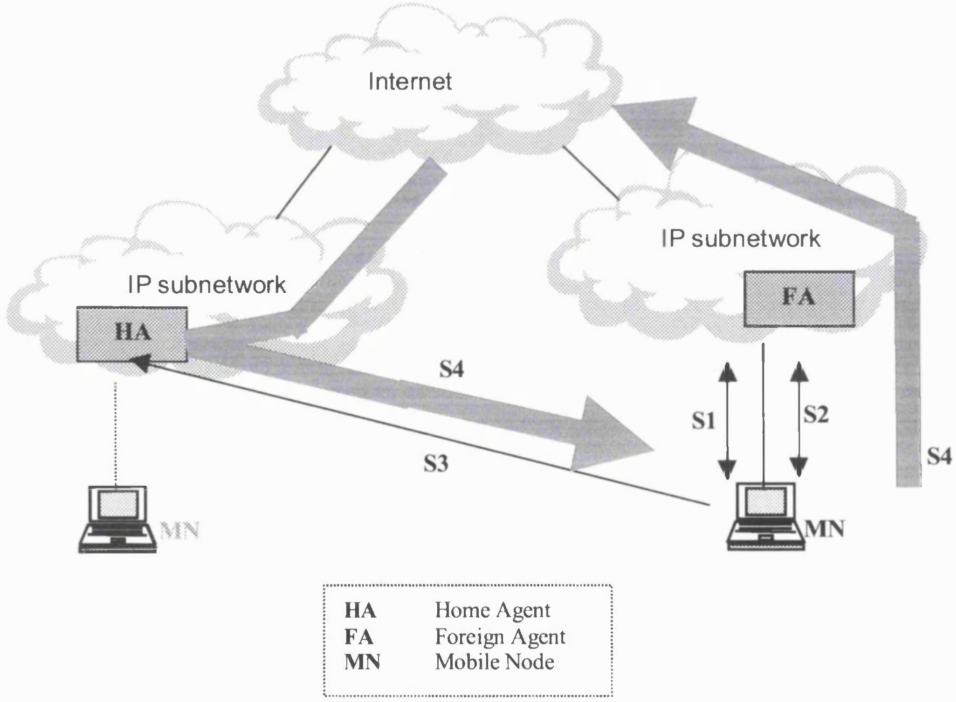


Figure 2-1: The operation of Mobile IP.

In the following paragraphs we discuss the Mobile IP approach and we point out the drawbacks of its architecture. These are related to a parochial perception of the location of a mobile node, the use of IP addresses for identifying mobile nodes and the limited scope of the Mobile IP architecture which makes it inappropriate for non-Internet environments or for addressing other kinds of mobility in communication networks.

2.1.1 Criticism

For the mobile node, its location is determined by the mobility agent advertisements. If it can receive advertisements from its HA, it understands that it is located in its home network. If the advertisements come from one or more FAs it understands that it is located away from its home network. The mobile node perceives the change of location by the change of the mobility agent that it is attached to. We can deduce that a mobile node associates its location with the mobility agents and the care-of-addresses that they assign to it. The mobile node perceives its own identity as an IP address, which it maintains regardless of its location.

The HA of a mobile node has the same perception of location and identity for the mobile node. The mobile node registers its location in the form of a care-of-address with a registration request to its HA.

However, for the other Internet nodes the location of a mobile node does not change. The HA offers a kind of location transparency to them. This happens because in the Internet, a node not only is identified by an IP address, but this IP address is also used for routing messages to it. As Paul Francis points out in his thesis [Francis94] an address could be split in two parts: the *identifier*, which identifies a network destination, and the *locator*, which assists in the routing function. In the Mobile IP architecture, the two parts of an address are distinguished by using the permanent IP address of a mobile node as an identifier, and the care-of-address that a mobile node is allocated when roaming as the locator [Francis94, RFC2002]. However, this distinction between identifier and locator for addressing a mobile node is only available to the HA of the mobile node. This distinction is not available to other nodes that may need to contact the mobile node. This is because the legacy of IPv4 does not accommodate such a distinction. This leads to many problems in the Mobile IP architecture, among which is the triangular routing problem that we discuss later.

It is this concept of the IP address and the IP legacy that makes the Mobile IP architecture inappropriate for addressing other kinds of mobility such as personal mobility or mobility of services. The Mobile IP architecture gives no answers to how persons or services can be uniquely identified. Also, it does not support separate identifiers to identify a mobile entity (such as person, service, or node) and its location. With the IP address space problem it is unlikely that IP addresses will be used to identify persons or services, and if they were used, we would still have problems with the inherent association between an IP-address-based identity and a location. Although such problems have been recognised earlier in the Internet community [RFC1498, Shoch78], they are still present in Mobile IP.

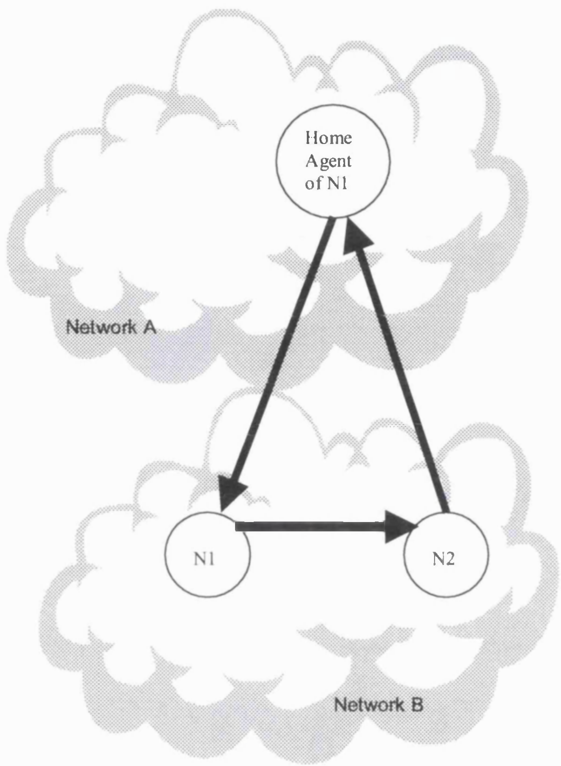


Figure 2-2: Illustration of a triangular routing situation.

Additionally, the triangular routing problem is another fundamental drawback of the IP architecture. For example, let us assume that a node (N1) of a network (A) moves to another network (B) and it communicates with another node (N2) in that network. The traffic from (N2) to (N1) will still have to be routed via the home agent of (N1) on network (A), despite the fact that both (N1) and (N2) are located on the same network (B). On the contrary, the traffic from N1 to N2 follows a direct route. This situation is known as *triangular routing* and is illustrated in Figure 2-2. We call this situation a problem because it can generate more network traffic than required by crossing different networks unnecessarily, it can cause asymmetry in the network performance for inbound and outbound traffic from the mobile node and in case that a link with the HA is lost, communication with the mobile node cannot be established although a direct route to the mobile node may be available.

In order to cope with the triangular routing problem a scheme for route optimisation in Mobile IP has been suggested [PJ00]. This scheme provides certain extensions to the Mobile IP protocol so that datagrams can be routed from a correspondent node to a mobile node without going to the home agent of the mobile node first. This is achieved by allowing correspondent nodes to cache binding information to mobile nodes so that they can tunnel datagrams directly to the care-of-address of mobile nodes. These extensions also allow for datagrams in flight when a mobile node moves and for datagrams sent to an out-of-date cached binding to be forwarded to the new care-of-address of the mobile node that they were destined.

The new version of the IP protocol, IP version 6 (IPv6) [RFC1883] offers a flexible scheme for IPv6 address allocation [RFC2073] and provides some mechanisms for maintaining a binding between an IP address and a care-of-address for a node, among other optimisations (a comparison of Mobile IPv4 and Mobile IPv6 can be found in [FS98]). This binding can be held by both the HA and other correspondent nodes. In this way, other nodes can communicate with a mobile node without tunnelling the traffic via the HA of the mobile node and thus the triangular routing situation is avoided.. There are already implementations of Mobile IPv6 [FS98].

However, although Mobile IPv6 solves some of the problems of Mobile IPv4 it still has some drawbacks. First of all, when a mobile node changes location, and in order to avoid the triangular routing problem, it will have to communicate its new care-of-address to every node that may contact it. This is an overhead that could be avoided by allowing the mobile node to communicate its new care-of-address to only one other entity (i.e. the home agent), from which other entities could obtain it.

Also the use of IPv6 addresses to identify a mobile node does not cure another problem that both the current IPv4 and the proposed IPv6 present: an Internet node can have more than one IP addresses and therefore, more than one identity. Because of this, every time a mobile node moves to another network (changes point of attachment to another network), it will have to register the IP address of each of its network interfaces separately. A single unique identifier for a node could make this procedure more efficient and speedier. That is because virtually, it is the mobile node that is to be contacted, not its network interfaces. It would also allow a HA to efficiently associate extra information in relation to its nodes, in the form of customisable node profiles. The use of profiles for mobile entities can enhance the quality of mobility support and they are particularly useful for other kinds of mobility such as personal mobility.

Finally, although it is stated that Mobile IP is to provide support for node mobility, the use cases that are supplied in the related IETF documents [RFC2002, PJ00] mention a user taking their laptop computer and attaching it to a different network, or a group of users with their laptop computers attached to the router of an aeroplane changing point of attachment to the Internet altogether. However, Mobile IP does not prescribe how a user can move to the terminal of an aeroplane maintaining the same environment that they have at their home terminal and it does not provide any architectural framework on how this could be implemented.

Also, Mobile IP does not address client/server mobility. In the case of server mobility, where a mobile server moves to another node, there is no provision for the clients of this server to stay attached to it or rediscover it. The use of the Domain Name System (DNS) [RFC1034] could provide a solution for the problem of locating a mobile server. However, the DNS was not originally designed for this purpose and it is recognised in the Internet community that DNS is not efficient enough for mobile servers since it can take a significant amount of time before a server location update reaches all the cached registers of the DNS.

Moreover, let us consider the example of a company relocating its network and maintaining the same IP address for its nodes and its routers. Mobile IP would have been inadequate in this case, since the problems of triangular routing would have had a very strong impact on network traffic and performance and, therefore, it would have been particularly inefficient to keep the old IP addresses. This justifies that Mobile IP not only cannot apply to other kinds of mobility but also that it cannot address the problem of node mobility in all its scope.

2.2 GSM

The Global System for Mobile telecommunications (GSM) by ETSI facilitates a number of architectural entities in its network architecture [GSM03.02]. For our purposes we outline a subset of those entities and their operation in a simple scenario. In reality there are more architectural entities involved in the scenario than those illustrated in Figure 2-3 and in Figure 2-4 [GSM09.02, GSM03.04, GSM03.12]. We believe however, that this simplification does not alter the main concept of the operation of GSM and that it suffices for the points that we want to make. The architectural entities that we employ in this scenario are:

E1. A *Mobile Station* (MS), which physically is the mobile terminal equipment with a Subscriber Identity Module (SIM) [GSM02.17] attached to it. Therefore, the term *mobile station* encapsulates this association between subscriber (user) and terminal. The mobile station can roam across different Public Land Mobile Networks (PLMN). A mobile station is identified by an IMSI number (International Mobile Subscriber Identity). The mobile terminal itself is identified by an IMEI number (International Mobile Equipment Identity) [GSM02.16]. Currently, the IMEI is used in GSM to identify stolen terminal equipment and it is used in conjunction with the IMSI. Both IMSI and IMEI are not related to the number that is dialled to call a mobile station. This number is called the Mobile Station International ISDN number (MSISDN), which has the format of the numbers that we dial today to call mobile telephones including the country and area code [GSM03.03].

E2. A *Home Location* Register (HLR), which is a database at the home network of the mobile station, i.e. the network where a mobile station is registered. The home location register maintains subscription, location and charging information for the mobile stations/subscribers. It also stores the IMSI and MSISDN numbers of the subscribers together with other subscriber related data [GSM03.08].

E3. A *Visitor Location Register* (VLR), which is a database maintained in networks with visiting mobile stations. It stores the IMSI and MSISDN numbers of the visiting mobile stations/subscribers. It also allocates a Mobile Station Roaming Number (MSRN) to each visiting station, which can be used to route a call to a mobile station at the visiting location. The VLR also stores supplementary parameters for the visiting mobile stations, which it obtains from their HLR.

E4. An *Equipment Identity Register* (EIR), which contains a classification of terminal equipment, based on the IMEI [GSM02.16]. The classification includes "white", "black" or "grey" lists and, based on them, a network can decide whether to accept calls from a mobile station that carries a specific IMEI. The IMEI is submitted by the mobile station together with the IMSI number to the network. Network entities can consult the EIR regarding this classification. The IMEI can be extended with information related to the software that a mobile terminal uses; in this case an IMEI Software Version Number (IMEISV) [GSM02.16] can be produced. Based on the IMEISV networks can decide whether a mobile station is able to use certain services. The IMEI is securely placed in the terminal equipment by the manufacturer [GSM02.16].

E5. A *Mobile-services Switching Centre* (MSC), which is an exchange that performs all the switching and signalling functions for mobile stations located in its area. The MSC also has to provide for the location registration and handover procedures for mobile stations [GSM03.02].

E6. A *Gateway MSC* (GMSC), which is a MSC that can also route calls to mobile stations. In order to do that, the GMSC has to interrogate the HLR of a mobile station and route the call to the appropriate MSC that this station is attached to [GSM03.02].

E7. A *Local Exchange* (LE), which is the exchange equipment that is used in a Public Switched Telephone Network (PSTN).

We can now look at a simple scenario, where a user (User 2) on the PSTN of one country (Country 2) calls another user (User 1) who uses a mobile station (i.e. a mobile telephone). The latter is registered in another country (Country 1) but when the call is made they are in the same country (Country 2), since (User 1) is visiting there. GSM will take a number of steps to deliver this call.

S1. When the mobile station of (User 1) moves to the network of the visiting country, it will have to submit its IMSI and the IMEI to the VLR of the visiting network in order to register. The MSC will mediate for this exchange.

S2. Based on the IMSI, the VLR will contact the HLR of the mobile station and it will inform it of the new location of the mobile station by means of an MSRN, which can be used to route calls to the mobile station at its new location. The VLR will submit the IMSI and the IMEI to the HLR to perform this operation.

S3. The HLR may perform a check with the EIR for the given IMEI before it provides the VLR with the subscriber/mobile station details. It will also have to update the

VLR of the network where the mobile station was previously, if that has been the case.

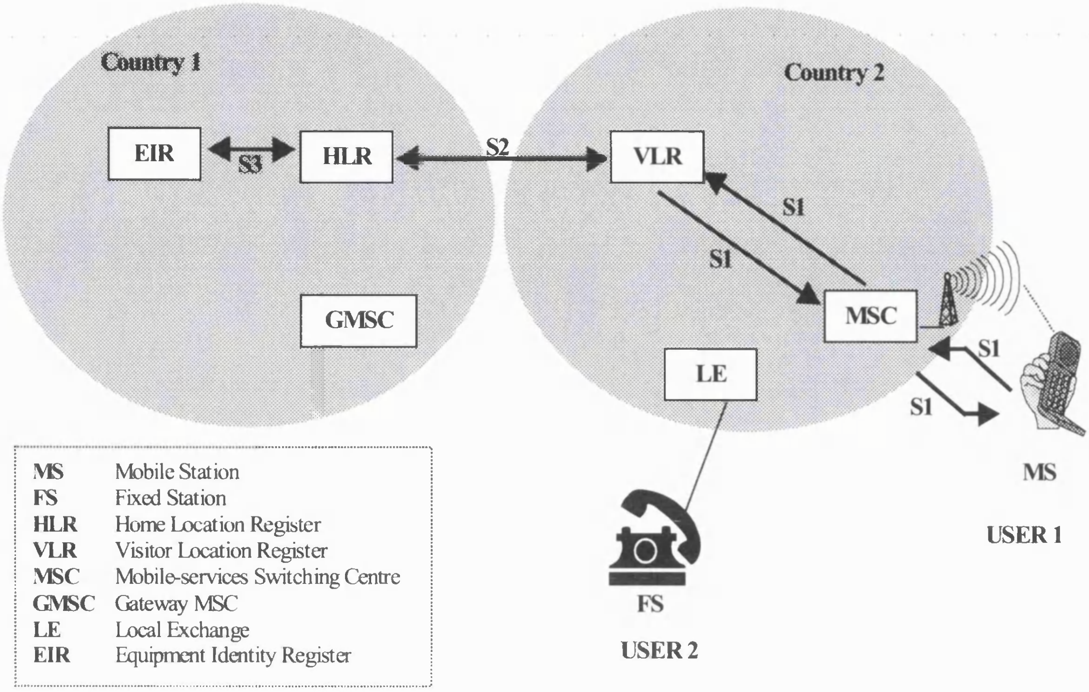


Figure 2-3: The operation of GSM – registration in a visiting country.

Steps 1-3 complete the registration procedure of the mobile station with the visiting network and they are illustrated in Figure 2-3. N.B. some GSM systems do not use IMEIs.

When (User 2) makes a call for (User 1) to the mobile station, the following steps take place:

S4. The user of the fixed telephone equipment (User 2) in (Country 2) will dial the MSISDN number of the called mobile station/subscriber. The LE that the fixed station is attached to will then go through the international exchanges to a GMSC for the called subscriber in (Country 1).

S5. The GMSC will query the HLR about the location of the subscriber with the given MSISDN (which maps to the subscriber IMSI). The HLR will return the MSRN to the GMSC.

S6. The GMSC can now route the call to the mobile station by using the MSRN. The call will be routed via the international exchanges through to the MSC that the mobile station is attached to and finally through to the mobile station of the subscriber.

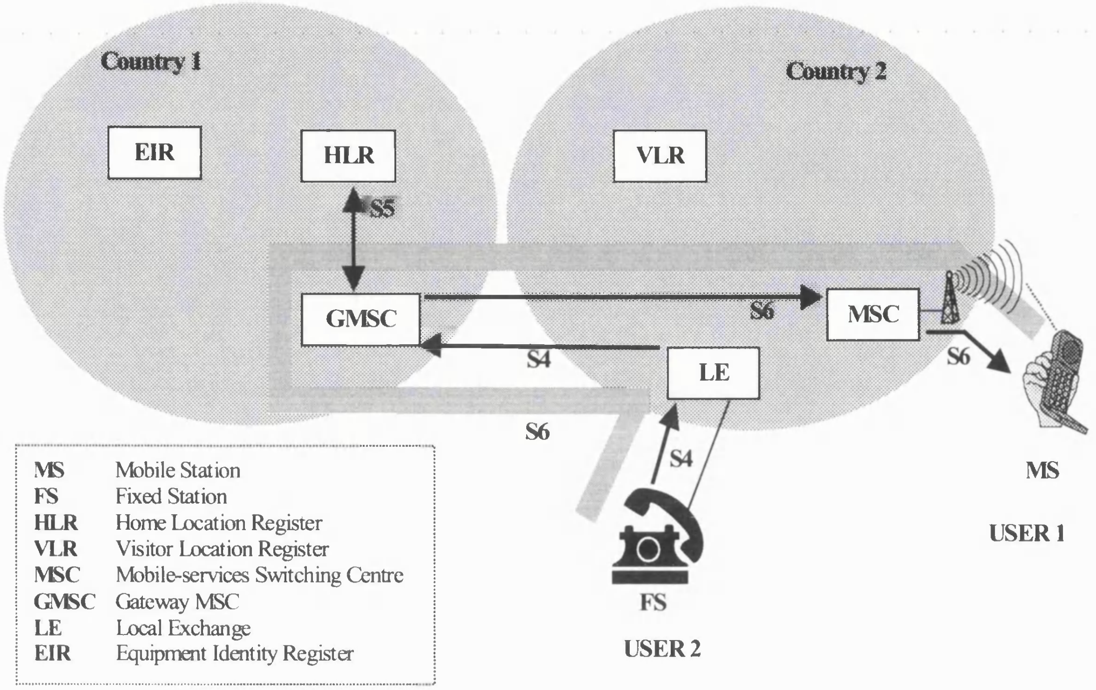


Figure 2-4: The operation of GSM – call to a mobile station in a visiting country and originating from the visiting country.

Steps 4 – 6 are illustrated in Figure 2-4.

2.2.1 Criticism

The legacy of traditional telephony is evident in GSM. The most important indication of this is the peculiar association between the users (or subscribers) and the terminal equipment that they use. In traditional telephony, a telephone number is used to address a user or a group of users. In reality it is used to address a piece of terminal equipment (the telephone) that is attached to a fixed location. Also, the telephone number is used to route a call to the particular terminal.

This association is inherited by GSM. The use of the International Mobile Station Identity (IMSI) number and its inherent coupling between a subscriber and the mobile station manifest this association. There are many drawbacks in this approach.

Firstly, it is possible to address the tuple "subscriber – terminal" but it is impossible to address either the subscriber or the terminal separately. In the GSM community this problem became evident when stolen mobile telephones (mobile terminals) had to be identified and when the capabilities of a mobile terminal had to be interrogated before it could receive some mobility services [Pand97, GSM02.16]. This led to the specification of an International Mobile Equipment Identity number (IMEI) for the first part of the problem, and the extension of the IMEI with a Software Version number field (IMEISV) for the second part. Still this approach has not been widely adopted yet [Pand97] and it is questionable if the software version field of the IMEISV can express the complex capabilities that will be required to support advanced mobile services in the future.

Apart from this, since a user (or subscriber) is bound to a mobile terminal, it is impossible to register and support more than one user on the same terminal. This problem is evident in developing countries, which having not very advanced wireline infrastructure, they invested in mobile telephony as a replacement to traditional telephony. In these countries, due to poor resources, it is not uncommon that a whole building is sharing one mobile telephone. Telephone terminals that support multiple numbers could provide a solution to this problem, provided that telephone numbers could be registered and deregistered dynamically on a telephone terminal.

Since a user cannot be addressed independently of the mobile terminal equipment, it is impossible to allow for personal mobility, where the users can use fixed terminal equipment and mobile terminal equipment interchangeably.

There is a very narrow scope to the concept of mobility in the GSM standards. With the association between a user and a terminal it is very unclear whether GSM provides for terminal mobility or personal mobility.

The answer to this is that GSM provides partly for terminal mobility and partly for personal mobility. Terminal mobility is not supported in a wide sense. For example, there is no provision for mobility of terminals that so far were considered “fixed” or for number portability. There is no provision for terminal equipment profiles, which presents a problem for offering advanced services where the terminal capabilities are important. Also, there is no consolidation of mobile telephony and traditional telephony; there is only interworking between them.

Personal mobility is supported in the way of allowing users to change geographical location freely, as long as they are hooked on a mobile terminal. On the other hand, the users' identity and profiles are intertwined with the information of their terminal equipment identity and capabilities. The Subscriber Identity Module (SIM) of GSM [GSM02.17] offers some degree of flexibility allowing users of the GSM service to use another mobile terminal for GSM access only. But the SIM itself has to hold service and terminal specific information (such as time for periodic location updating, update status, forbidden PLMNs). The SIM also uses the IMSI to identify users and is supposed to be attached to the terminal on a semi-permanent or permanent basis [GSM02.17]. The ITU standard for Universal Personal Telephony (UPT) [F.850] has proposed a scheme that allows users keep a unique identity regardless of the equipment that they use. However, these proposals have not been adopted by GSM.

The traditional telephony influence is also evident in another drawback of GSM. In the scenario that we describe in the previous section, we see that the connection that is established between the mobile and the fixed terminals crosses the country boundaries twice, although both terminals are located in the same country in that instance. This triangular routing problem cannot be easily overcome in GSM. Although GSM allows for a LE directly to contact the HLR of the called mobile station before it establishes the connection, this approach is not recommended. In the GSM recommendation 03.04 [GSM03.04] it is stated explicitly that this approach would complicate the process of charging the called subscriber for the service, if the subscriber is roaming to a foreign network or if the call is to be forwarded.

From this analysis we see that the GSM architecture uses a confusing concept of mobility. This is because it provides for partly personal mobility and partly terminal mobility as discussed above. GSM has a narrow scope for terminal mobility since it mainly provides for terminals with telephony capabilities and which can have only one user attached to them. The concept of personal mobility is parochial too, since users are associated with mobile terminals. Based on the traditional telephony legacy and with the initial objective of offering a commercially operational service immediately, GSM is not expected to provide the framework for third-generation mobility and the advanced services that come with it.

2.3 Other Telecom standards

ETSI has recognised that GSM is inadequate for third-generation services and they have focused their standardisation efforts on the Universal Mobile Telecommunications System (UMTS) standard. Still, UMTS has not produced complete recommendations so far. ETSI has also been following the ITU developments on the requirements for third-generation mobility and they have identified several requirements, among which, requirements for the management of mobile services [UMTSmgmt] and for the deployment of satellite services [UMTSsat]. Their approach on management is TMN-based [M.3010] and is looking at how to integrate existing management frameworks with UMTS.

ITU is aspiring to complete very soon the new IMT-2000 standard (International Mobile Telecommunications by the year 2000) [M.687-2, BFMC97, PGLM97]. IMT-2000 is based on the previous FPLMTS (Future Public Land Mobile Telecommunication System) [M.687-1]. At the moment of writing there are no concrete results yet but from the output that has been published [BFMC97, PGLM97] it seems that IMT-2000 aims to integrate and consolidate existing ETSI standards (like the GSM) and ideas from FPLMTS. They also aim to provide for UPT over both fixed and mobile terminals.

The IMT-2000 proposals divide the users in two classes. One class is the UPT subscribers that can use any fixed or mobile terminal by registering their UPT number with the network [PGLM97]. The other class is the mobile telephony subscribers that can be provided with some form of personal mobility by means of a SIM card. For those users, the binding between user and terminal equipment is maintained as with GSM today. For example, the IMSI and IMEI are maintained with no significant changes [PGLM97]. The aim of IMT-2000 is to provide for full commercial deployment of mobile services and to accommodate the commercial base of the current mobile telephony subscribers and standards at the same time. With this approach, IMT-2000 may eventually inherit the drawbacks of the existing GSM architecture and it may not be able to address mobility requirements in a wider scope.

2.4 Related work

2.4.1 ODP and multimedia systems

Nigel Davies has investigated the impact of mobility on Distributed Systems platforms from the QoS side [Davies96]. Also, suggestions for distributed systems platforms that support applications adaptive to the QoS requirements of mobile environments have been investigated [DFWB98]. The approach of this thesis does not address QoS requirements for mobility. Extensions to the FoT in order to support QoS attributes are suggested in the further work section.

2.4.2 The OSI reference model

As explained in the introduction of this thesis, the Framework of Terms (FoT) in this thesis is orthogonal to the Open Systems Interconnection Reference Model (OSI-RM) [OSI-RM]. This work is mostly in line with the work by Saltzer on requirements for the identification of services, users, nodes, network attachment points and paths in data communications networks [RFC1498, Shoch78]. One difference is that we address these concepts at a more abstract level. For example, the term network entity is abstract to describe any kind of entity that can be considered virtually or physically attached to a communications network. Also, in this thesis it is justified why and under what circumstances unique identifiers for network entities are required and the scope of uniqueness for these identifiers is specified. Finally, the scope of this work is not confined to the investigation of identities for fixed and mobile entities.

2.4.3 Addressing and locating network entities

In relation to the work by Paul Francis, the concepts used in this thesis are similar to the "identifier" and the "locator" [Francis94] for addressing a network entity. The requirements so that an identifier can unambiguously identify a network entity are investigated and the necessity of identifiers is speculated on. Also, the requirements for unambiguously locating a mobile network entity are examined. However, this thesis is not concerned with addressing as such and a communications environment is considered independent of network technologies or protocols. The data transport between the network entities of our framework is taken for granted and communications protocols and addressing are out of the scope of this work.

2.4.4 Formal models for mobility

In this section we present current work on formality for mobility. This background comes from the community of concurrent distributed systems and parallel programming. We present calculi for modelling mobile systems and for investigating the equivalence between mobile system specification and design (*π-calculus*). We also present tools for modelling hierarchical administrative domains within which mobile processes can be securely contained (*ambient calculus*). Finally, we present how mobile systems that can experience disconnection or failure can be expressed and verified formally (*mobile UNITY*).

How the framework of terms of this thesis relates to these is provided at the end of chapter three.

2.4.4.1 CCS and the π-calculus

The π-calculus [MPW92] is an extension of the process algebra called *CCS* (*Calculus of Communicating Systems*) [Milner89] or *Process Calculus*.

The CCS algebra is the contribution from theoretical computer science to understanding and modelling concurrent communication systems. The basic assumption in CCS is that there is an element of concurrency to a system since it is composed of a number of *independent*, *concurrently acting* parts, each part with its own identity. These parts achieve a unity within a system via *communication* between them.

The different parts of a system are called *agents* in CCS. These agents communicate with each other by taking part in indivisible acts of communication that are experienced simultaneously by both a sender agent (where a message originates) and a receiver agent (where a message is destined). These indivisible acts of communication are called *handshakes*.

CCS provides the means for describing communicating systems precisely and it also provides a set of equational laws. With these laws, the equivalence between two agents can be investigated. Strong equivalence or *strong bisimulation* between agents is based on the idea that we only distinguish between two agents if the distinction can be detected by an external agent interacting with each one of them. For strong equivalence every action of one agent must be matched with an action by the other agent, even their internal actions. On the other hand, weak (or observation) equivalence or *weak bisimulation* relaxes the requirement that even the internal actions of two agents have to be matched; one internal action in one agent can be matched with zero or more internal actions by the other agent. Internal actions of agents do not represent a potential communication. For this reason, they are not directly observable.

Every agent can have two kinds of ports for handshakes: *input ports* and *output ports*. For example, in CCS notation the input port *in* on which an agent C can receive a handshake is represented by *in(x)*. In this case, the handshake becomes the value of the variable *x*. Similarly, C can later output value *x* to output port *out*. CCS provides a notation for describing constants, variables and expressions that can in this way describe an agent's behaviour. Other agents that know the name of the input port *in* of C possess a link *in* to C, which they can use for handshakes with C. For example, an agent C′ can transmit a value *4* to agent C through the input link *in* of agent C. The value *4* will then become the value of *x*.

The CCS is asynchronous. However, there is a synchronous version of the CCS. The idea in the synchronous CCS is that concurrently running processes proceed in lockstep, each performing one action at each time instant. The asynchronous CCS can be expressed in terms of the synchronous CCS.

The extension of the π-calculus gains simplicity over CCS by removing the distinction among link names, variables and data values. Instead all these are called *names*. A system becomes a collection of agents that can exchange handshakes between them passing names to each other. Therefore, there are two kinds of entities in the π-calculus: *names* and *agents* or *processes*. In this way, names of links can be exchanged between agents. Assume that an agent P needs to transmit a value *4* to agent Q. The π-calculus allows P to delegate this task to another agent R. All that P has to do in order to delegate this task to R is to transmit to R the name of the link to Q and the value *4* to be transmitted along this link.

By allowing an agent P to acquire a link to another agent Q and to give up a link to a previous agent Q′, a kind of "mobility" can be achieved: agent P can "move" by moving its attachment from agent Q′ to another agent Q.

Since the names of links are not necessarily unique among all agents, there is the case of *scope intrusion*. This happens when an agent is passed a name of link to an agent, which already possesses another link with the same name. This case can be dealt with by renaming links so that a uniqueness of name of a link can be preserved within the scope of these names. There is also the case of *scope extrusion*, when an agent passes a private link to another agent.

Finally, by not distinguishing between constants and variables, when a name substitution occurs in the π-calculus strong bisimilarity is lost. For this reason, the π-calculus introduces the concept of *strong (non-ground) equivalence*, which allows all name substitutions [MPW92]. Similarly, a concept of a *weak (non-ground) equivalence* is investigated [MPW92].

Different flavours of the π-calculus have been defined and employed for different purposes. The differences between these π-calculi are either minor notation differences or more important changes justified by particular theoretical research or application [Sewell99]. Among these π-calculi, there is the *polyadic π-calculus* [Milner91] that makes the encoding of polyadic input and output from agents more flexible. Also, there is the *Spi* calculus [AG97] for cryptographic protocols, the *ambient calculus* [CG98] for expressing security domains, which is presented in the following section, and others. There is also a distinction between *higher-order* and *first-order π-calculi*; the former allow agent-passing, while the latter follow strictly the name-passing paradigm [Sang96].

Also, different kinds of bisimulation have been identified. Among these there is the concept of *open bisimilarity* [Sang93] and the *labelled bisimilarity* [Sang98]. Each of the different kinds is used for different purposes and it is not clear which one should be generally preferred. It must be noted however that most bisimilarities are not congruence relations [Sang96].

A Mobility Work Bench (MWB) tool has been deployed for describing open bisimulation equivalences for mobile concurrent systems described in the π-calculus [VM94]. Using this tool the equivalence of the formal specification of the handover protocol of GSM has been verified against its service specification [VM94].

Also, a number of programming languages has been designed or proposed for the π-calculus, among which is *Pict* [PT97].

2.4.4.2 Mobile Ambients

Starting from the work that has been done for the π-calculus, the concept of mobile ambients [CG98, CG97] and the corresponding *ambient calculus* address issues that could not be captured with the standard π-calculi. These issues concern the definition of *administrative domains* within which agents are confined, administrative domain *hierarchies* and the ability of agents to move in and out of such domains in a secure way. The inspiration for this work came from the realisation of the potential for mobile computation over the World-Wide Web (WWW) [CG98], where firewall restrictions apply and can present obstacles to mobile computation. By providing a model for secure mobile computation over the WWW such restrictions could be applied in a proper way.

The crucial concept that is introduced is that of an *ambient.* An ambient is a *bounded* place in which computation can take place. An ambient can contain agents (*processes*) or other ambients (*sub-ambients*). An example of an ambient is a virtual address space. Example of agents contained in such an ambient are processes or data objects. The agents contained in an ambient can control the ambient. Each ambient has a *name. The name of an ambient can be used to control access to it* and therefore, it should be guarded.

The control of an ambient can be decomposed to a set of *capabilities*. Capabilities can be extracted from the name of an ambient but not vice-versa. Capabilities, once given to other ambients, allow these ambients to perform certain operations on it without knowing its true name, and therefore, without having full control of it.

There are three kinds of capabilities that an ambient can give out to other ambients. The *entry* capability allows an ambient to enter another one. Similarly, the *exit* capability allows an ambient to exit another one, while the *open* capability allows an ambient to open up another ambient and release its content.

Communication mechanisms are considered orthogonal to the mobility primitives of mobile ambients. However, it is assumed that some primitive form of communication that is local in an ambient can exist, while messages across ambients should be restricted by capabilities. A number of I/O primitives have been defined for mobile ambients [CG98].

Given these I/O primitives, the mobile ambients calculus can express the asynchronous π-calculus. In order to do this, a channel of the π-calculus can be represented by an ambient. The name of the channel can become the name of the ambient, while the communication on a channel can map to local communication inside an ambient. Communication between two remote ambients can be modelled by the movement of "messenger" agents between them; these messenger agents should possess the capabilities of entry and exit of all the ambients (administrative domains) that will be encountered on the way.

The ambient calculus work introduces a new kind of equivalence, the *contextual equivalence* [GC99]. Two processes are contextually equivalent "if and only if they admit the same elementary observations whenever they are inserted inside any arbitrary enclosing process" [GC99].

Comparing to the π-calculus, while the π-calculus groups processes (agents) in a single, contiguous centralised collection (the system), the ambient calculus groups processes in many, separate distributed ambients. Apart from this, while the π-calculus allows the exchange of names between processes, the ambient calculus allows exchanges between entities only when they are located in the same ambient. Finally, the ambient calculus introduces access control by means of capabilities, while the π-calculus does not employ any such concept.

2.4.4.3 Modelling mobility using Mobile UNITY

Mobile UNITY presents a modular *notation* for expressing mobile computations and *logic for reasoning* about their temporal properties [RM98]. It extends the UNITY [CM88] model of concurrency with additions for mobility. These additions come from the realisation that mobile systems have to be *decoupled* and *opportunistic.* The former means that applications should be able to work with weak connectivity or no connectivity at all. The latter means that interactions should be accomplished whenever connectivity is available.

In mobile UNITY a *system* is a collection of independently executing *components (programs)*, which can *migrate over a space* (logical or physical). This movement can change the pattern of connectivity among them [RM98]. The concept of *proximity* is also introduced. When components are in close proximity mobile UNITY can express data sharing between them (*transient variable sharing*) or synchronisation of their actions (*transient action synchronisation*) [RMP97]. The state of a *program* is enhanced with the concept of *location* whose change of value represents motion [RMP97].

The goal of the mobile UNITY work is to characterise fundamental issues facing mobile computing while the pragmatic objective is to develop techniques for the verification and design of dependable mobile systems. The main area of application is that of *ad-hoc* networks, on which the proof logic of UNITY is employed for reasoning about mobile computations [RMP97].

The lack of a concept of space in the π-calculus, which is the other formal model of concurrency to compare mobile UNITY against, leads to inefficiency for expressing disconnection of mobile entities or failure. *Space* can be expressed in mobile UNITY. The *position* of a component can be expressed in terms of a variable that can be allocated values from a designated space.

An example of using mobile UNITY [RMP97] is that of modelling a system of carts that can move along tracks. On the sides of these tracks there are stations where the carts can load or unload goods. The movement of the carts, their position and their operation can be modelled in mobile UNITY. Collision avoidance can be controlled with action synchronisation and variable sharing (e.g. sharing of the variable that contains each other's position). By using the UNITY proof logic, collision avoidance in the specified system can be formally verified. For such a system, mobile UNITY can provide the means for describing it and verify its correct operation formally.

Mobile UNITY has already been applied for the verification of the *Mobile IP protocol* [MR99]. In this case, failure and disconnection of mobile nodes has been modelled successfully. Mobile UNITY has also been successfully used for modelling *code mobility* [PRM97].

2.4.5 Other related work

Gordon Blair and Jean-Bernard Stefani have successfully used an approach similar to the one of this thesis to investigate requirements for multimedia in open distributed systems, in their book "Open Distributed Processing and Multimedia" [BS98].

Technology viewpoint solutions for mobile computing have been investigated by Chander Dhawan in his book "Mobile Computing: A systems integrator's handbook" [Dhawan97]. In this thesis, the technology viewpoint is not addressed.

3. A Framework of Terms (FoT) for Mobile Systems

In this chapter we introduce a framework of terms that provides a terminology for mobile systems on an abstract level. We give an abstract definition of a *communications network*, which comprises *network entities* that are interconnected with *media*. We define terms like *entity location*, *message exchange* between entities and we attempt a *definition of mobility*. This Framework of Terms (FoT) is expected to provide the basis for exchanging ideas and experience on the deployment of mobile systems. It is also expected to provide a common understanding of mobile systems by introducing a technology independent terminology. The terms of the FoT are defined informally and in plain English so that they can be understood by mobile systems designers coming from different backgrounds.

In the definitions of the FoT we distinguish between fixed and mobile entities. This might be seen as contradictory to our initial statement (in chapter one) that mobility should be considered in the design of communication systems from the very beginning as a default property of any system. However, our intention for distinguishing between fixed and mobile entities is different. It is to investigate how and why mobile entities were not considered in legacy systems and what are the properties that make an entity mobile and not fixed. Realising what the difference between fixed and mobile entities is could help for building better systems in the future.

Apart from introducing a new terminology, in this chapter we also investigate issues related to the *identification* of fixed network entities. The mobile entity identification requirements are presented in the next chapter, where we investigate the requirements for mobility from the information and computational viewpoints using the FoT.

The FoT provides technology independent terminology for mobile systems. The terms are defined in an informal way, in plain English and they are accompanied by a number of examples. As a result of the informal nature of the FoT definitions the terms of the FoT may be ambiguous. In order to reduce this ambiguity, at the end of this chapter we describe how the FoT can relate to formal models for mobility and we can express some of its terms using a formal notation (π-calculus).

3.1 Communications network

Every communications environment can be perceived differently at different levels of abstraction. We believe that the basic characteristics of a communications environment are the same though, regardless of the level of abstraction that we are referring to. We claim that a communications environment of any level can be perceived as a mechanism of message exchange among entities via some media.

The first step in describing a communications environment is to identify the entities and the media of this environment. The entities and media that we identify depend on the viewpoint that we use to look at the communications environment.

Although this statement looks very similar to the concept of the OSI-RM, there is not necessarily a direct mapping between the OSI-RM layers and our levels of perception of a communications environment. For example, if we were investigating mobility on the network layer of the OSI-RM, we could model a number of network layer peers as network entities, and their connections over a network layer protocol as the media. On the other hand, we could model a person and a terminal as network entities, where the person is attached to the terminal via the input/output interfaces that the terminal offers; these interfaces would be the media.

For example, let us assume a home user, who is accessing a specific web page on the Internet. The user has Internet access via an Internet Service Provider (ISP), to whom he connects over a dial-up connection, as in Figure 3-1(a). From the user's view, the communications environment includes his home terminal, the host of the ISP that he is connecting to and a number of web-sites that he can access via his ISP; these are the entities of his perception of a communications environment. Also, this environment includes the dial-up connection to the ISP and the connection from the ISP to the web-sites. If any of these connections is down connectivity is impossible; these connections are the media of the user's perception of a communications environment.

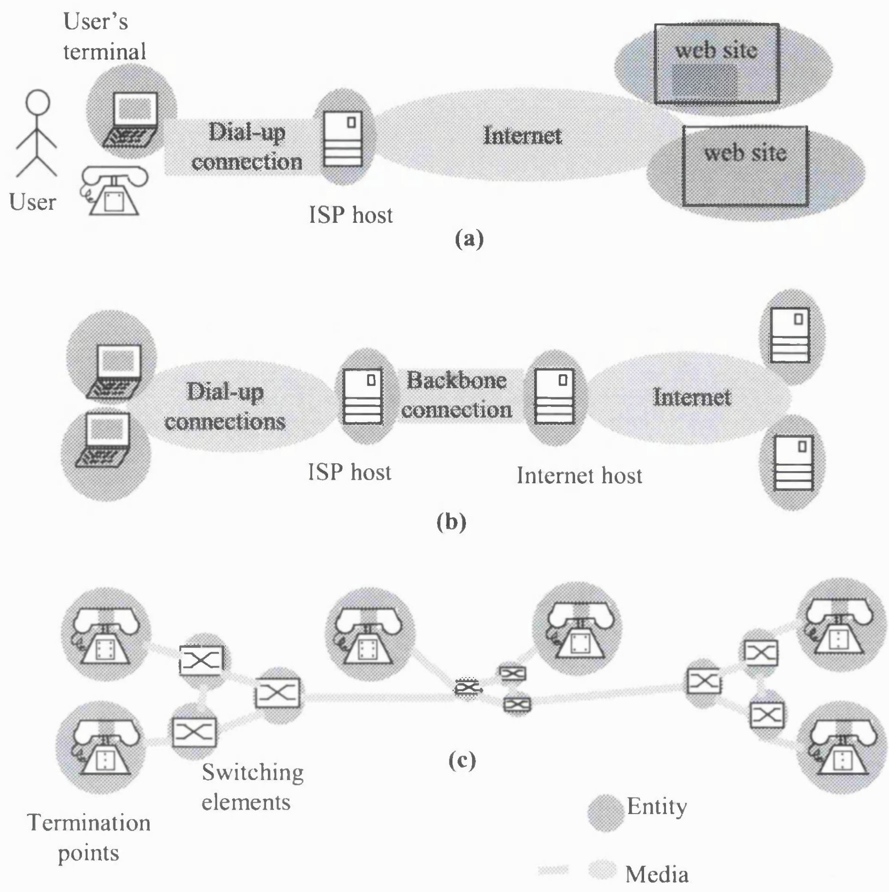


Figure 3-1: Three different communications networks as perceived in the same communications environment: (a) home user's view, (b) Internet Service Provider's view, (c) Telecom provider's view.

In the same example, the ISP view of the communications environment differs from that of the home user. The environment has a number of home users whose terminals are additional entities that gain connectivity to some of the hosts that the ISP owns. The users' hosts use the dial-up media to connect to the ISP ones. Also, the ISP connects to an Internet backbone via another host, which is another entity, and the connection to this entity is an additional medium. The ISP's perception of the communications environment is illustrated in Figure 3-1(b).

We can also consider the view of the telecom provider who offers the links between the home users' terminals and the ISP host. According to this view, which is illustrated in Figure 3-1(c), the communications environment consists of a number of terminal points to which the home users' terminals and the ISP hosts are attached and a number of switching elements; these are the entities. The media are the connections between these entities that allow message exchanges between terminal points. Figure 3-1 illustrates these three different levels of perception of a communications environment. Each perception of this communications environment features different entities and media. We understand that in real systems, the view of the ISP or the Telecom provider can be more complex and involve many more entities than those of our example. We believe, however, that this simplified view serves as well for the purpose of our example.

As this example illustrates, a communications environment consists of a number of entities that are interconnected by media. Also, there can be more than one view or perception of a communications environment. Each of these views or perceptions may feature different entities or media.

> ***Communications Environment:*** *an environment that features a number of entities and media. The entities and the media can be perceived at different levels of abstraction, which offer different views of the environment. Each view may feature a different set of entities and media.*

Each view of a communications environment we call a *communications network*. A communications network is composed of a number of *elements*. We distinguish between two kinds of elements: *network entities* (or *entities*) and *media*. We also assume that a message exchange between network entities of a communications network signifies a *service* offered by one entity to another.

The *smallest* communications network, in terms of number of entities and media involved, is one that features *one medium*, to which *two entities* are attached. The case where there is no medium in a communications network is invalid in our framework, since it implies that if there are network entities, they cannot exchange information between them, and therefore, there is no network. The case where only one entity is involved in a communications network is also invalid in our framework, since our concept of a communications network provides for exchange of information between different entities.

> ***Communications Network or Network:*** *a particular view of a communications environment. It is composed of two kinds of "elements": "network entities" (or entities) and "media". The entities can exchange "messages" between them via the media. Each message exchange signifies a "service" that one entity can offer to another. A minimum communications network involves one medium and two entities attached to it.*

We can consider numerous other examples to which this concept of a communications network can map. Apart from data communications networks, where the network entities are physically attached to each other via wireless or wireline media, we could also describe an organisation as a communications environment. From the viewpoint of a manager for example, there is a communications network in this organisation, where the entities are the employees and the media that they are interconnected with are the telephone service, the e-mail, or the contact between them when they are in the same office.

Similarly, the radio link that the taxis use in some big cities can be perceived as part of a communications network from the taxi drivers' viewpoint. This network has a number of network entities, which are the radio-equipped taxis and the centre that broadcasts the taxi requests over the radio link. The medium of this network is the radio link to which both the centre and the taxis are attached.

Having described the concept of a communications network we will now define the concept of an entity and the concept of a medium in more detail.

3.2 Network entity (NE)

A *network entity* is the part of a communications network, which can participate in a message exchange as a source or sink of information. The media of a communications network are just the conveyors of this information. For example, the ISP host in network (a) of Figure 3-1 is a NE because the home-user can exchange authentication information with it in order to gain access to the Internet. The web pages are also NEs because the home user can receive information from them.

> ***Network Entity or Entity:*** *is the part of a communications network that can participate in message exchanges as the source or sink of information. The same physical entity of a communications environment can be perceived as a different network entity in different views of a communications network.*

A NE needs to attach or be attached to a number of media in order to participate in a message exchange. In a communications environment, a network element can be perceived as a network entity in one view of a communications network, while it can be perceived as part of a medium in another. For example, in communications network (b) of Figure 3-1 the NE that connects the ISP host to the Internet backbone is perceived as part of the medium that connects the ISP host to the Internet web sites in communications network (a).

As an example, persons, switching elements, software applications or terminals could be considered network entities in some communications networks.

3.3 Medium

In a communications network, a *medium* is the element that conveys information among entities that are attached to it. We can distinguish between two kinds of media:

M1. Unicast media

M2. Broadcast media

> ***Medium:*** *is the part of a communications network that can convey information among network entities that are attached to it. In this way, network entities can have message exchanges. A medium cannot be the source or sink of information; it only serves as the conveyor of information. There can be unicast or broadcast media.*

A unicast medium allows a network entity to convey information to a specified other entity to which it is attached at a given time. A broadcast medium allows a network entity to convey information to all the entities that are attached to it, at that given time. A broadcast medium cannot address particular entities.

There is also the case of multicasting, where it is possible to address all the entities that are members of a multicast group in a single exchange. We consider multicasting a special case of broadcasting. In our framework each multicast group is a separate broadcast medium, which can be either statically or dynamically created.

For example, let us assume that we have a number of terminals connected to each other with an Ethernet. This Ethernet link can serve as a number of different media according to the use that we make of it:

M1. A unicast medium.

M2. A broadcast medium.

M3. Many broadcast media (one for each multicast group that can be accommodated).

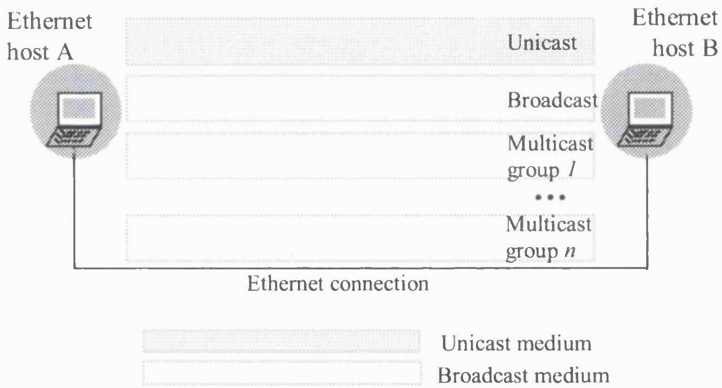


Figure 3-2: Different media between two hosts with an Ethernet physical connection.

In this example, two hosts that belong to different multicast groups can be seen as attached to different media, as illustrated in Figure 3-2.

The concept of a medium has a wide scope. In the example of the organisation of section 3.1, the personal contact of employees located in the same office can be considered a unicast medium. Similarly, a conference between employees in a meeting room can be identified as a broadcast medium that associates the group of employees that participate in the conference.

In order to demonstrate the scope of the concept of the medium let us consider the following example. Assume there is a network entity (A) that is connected to a number of other network entities via a number of media, but every time it needs to contact any of them it broadcasts the message to all of them via all the media that it is attached to. In this example, we should consider that this network entity (A) is connected to the other network entities by *a single broadcast medium*. The definition of a unicast medium is to convey a message to a *specified* network entity. In this case, entity (A) does not need to specify the entity to contact; therefore, entity (A) makes use of a broadcast medium, not a unicast one.

3.4 Communications network topology (CNT)

The *communications network topology* is defined by the arrangement of the entities and the media of a communications network at a given point in time. The arrangement of entities and media in a network can be *static* or *dynamic*. In the case of a static arrangement, the CNT does not change during the course of time. In the case of a dynamic arrangement, the CNT can change over time. Most networks need to change topology in order to fix faults, to improve the network performance, or to accommodate mobile entities.

Our example communications network with the employees of an organisation can be considered to have a dynamic topology. Every time an employee joins or leaves the organisation the arrangement of the entities changes, in the sense that more entities are added or removed from the communications network. Also, when a conference takes place, a new medium is established and a number of entities attach to it.

> ***Communications Network Topology:*** *the arrangement of network entities and media in a communications network. This arrangement can be static or dynamic.*

3.5 Communications path (CP)

A *communications path* is the arrangement of media and entities that is formed in order to convey a message exchange between two entities. These two entities are the *communications path end-points* (CPEP). The CPEPs are not part of the communications path. A CP can include a finite number (n) of media and ($n-1$) network entities, where ($n=1,2,3,...$). The shortest CP comprises a single medium.

***Communications Path:** an ordered list of media and entities that is formed in order to convey a message exchange between two entities. The communications path does not include the entity that originates the message exchange or the entity to which the message exchange is destined.*

We are going to show how the concept of a CP applies to number of communications networks with different topologies. In Figure 3-3 we see example CPs between two entities in a number of different topologies. The CP exists only during an exchange between two entities (the CPEPs). Other exchanges between the same entities may result in the creation of a different CP.

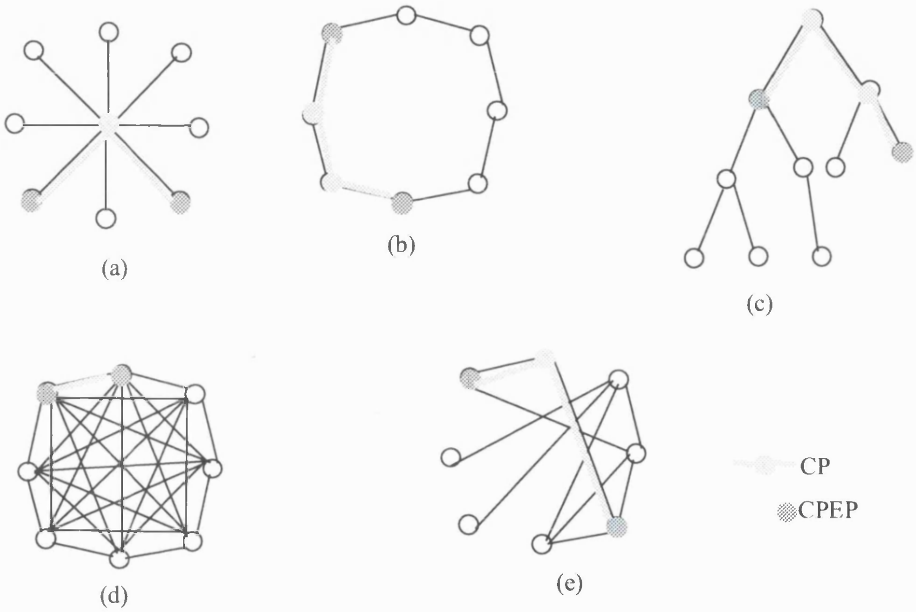


Figure 3-3: Example of CPs between network entities in different point-to-point topologies: (a) Star, (b) Ring, (c) Tree, (d) Complete, (e) Irregular.

We only consider CPs with two CPEPs. Cases where information is conveyed collectively to a group of entities can be fragmented to a number of different CPs with two CPEPs each. For example, the exchange of one home user who is having a multicast session with a number of other users of a multicast group can be defined as a number of CPs, one for each message exchange between the home user and each other user of the multicast group.

There are networks where more than one copy of the same message can arrive from a source entity to a sink entity or networks where a message exchange can be conveyed to one entity out of a group of entities; for example *anycast* networks. In this case we consider that the CP includes the arrangement of media and entities that conveyed the first copy of the message to the sink entity.

In a connection-oriented environment a CP between two entities is determined before the message exchange takes place, but this is not the case for connection-less networks. In connection-less networks the CP is not pre-determined because different messages can follow different routes to reach their destination. For connection-less environments, we consider different CPs for each single message exchange between two network entities. For example, in IP networks we consider a separate message exchange for each IP packet that is routed from a source node to a destination node.

3.6 Message exchange (exchange)

As we mention in the previous section, a CP can be formed in order to accommodate a message exchange between network entities. We also mention that a message exchange originates from a source entity and that it is destined to a sink entity. In this paragraph we define a message exchange in a way that is similar to the unconfirmed service elements of the OSI-RM [OSI-RM].

> ***Message Exchange or Exchange:*** *an asynchronous, one-way exchange of a message between two network entities. The entity from which the message originates is the source entity. The entity to which the message is destined is the sink entity. During a message exchange, the sink entity can be seen as providing a service to the source entity via this message exchange.*

A two-way message exchange is a special case of a combination of two one-way message exchanges.

A three-way exchange is a special case of a combination of three one-way message exchanges. Similarly, n-way exchanges are special cases of one-way message exchange combinations.

For two-way message exchanges, different communication paths can be facilitated for the first and the second part of the exchange. Similarly, for n-way message exchanges, different communication paths can be facilitated for the n different parts of the message exchange.

3.7 Entity location (location)

Let us assume a network entity (A), which is to have a message exchange with another network entity (B). When the exchange takes place, a CP is arranged with a number of media and entities, which support the exchange between (A) and (B). We consider that the CP that is set up to convey this message exchange determines the location of entity (B). According to the CP that is arranged, entity (A) can see entity (B) as either a *local entity* or a *remote entity*, as illustrated in Figure 3-4.

3.7.1 Local network entity (local entity)

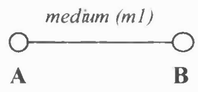
A network entity (B) is *local* to a network entity (A) when the CP that supports a message exchange between them consists of only one medium. The *location* of the local entity (B) is described by the medium that connects entity (A) to entity (B), as shown in Figure 3-4(a). Note that in our example entity (A) is the source entity of the message exchange, while entity (B) is the sink entity. The location of the sink entity, as it is perceived by the source entity, is described by the medium that connects the source entity to the sink entity. Therefore:

> *location*$_{source\ entity}$*(<local sink network entity>)=<medium that is used to connect the source entity to the sink entity>*

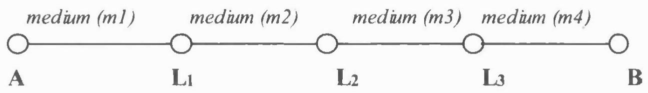
From this example we can see that the location of a network entity can be determined when this entity is contacted by another network entity via a message exchange. Also, we define the sink entity's location, as it is perceived by the source entity. Therefore, during a message exchange the location of the sink entity is *relative to the source entity.*

3.7.2 Remote network entity (remote entity)

A network entity (B) is *remote* to a network entity (A) when the CP that supports a message exchange between them includes at least one other entity.



(a) Entity (B) is local to entity (A)
The location of (B) is (m1).



(b) Entity (B) is remote to entity (A)
The location of (B) is (L_3).

Figure 3-4: The location of an entity (B) when another entity (A) initiates a message exchange with it: (a) entity (B) is local to entity (A); (b) entity (B) is remote to entity (A).

The *location* of the remote entity (B) is described by the last entity encountered in the CP from entity (A) to entity (B). For example, in Figure 3-4(b), the location of entity (B) as it appears to entity (A) is described by entity (L_3). Note that in our example entity (A) is the source entity of the message exchange, while entity (B) is the sink entity. The location of the sink entity, as it is perceived by the source entity, is described by the last entity on the communications path that connects the source entity to the sink entity. Therefore:

location$_{source\ entity}$*(<remote network entity>)=<last network entity on the CP from source to remote entity>*

From this example we can see that the location of a network entity can be determined when this entity is contacted by another network entity via a message exchange. Also, we define the sink entity's location, as it is perceived by the source entity. Therefore, during a message exchange the location of a remote sink entity is *relative to the source entity.*

In the case of a multihomed entity, i.e. an entity that can be reached directly via more than one other entity, its location for a particular message exchange is determined by the first network entity to pass the message exchange to it. This is another factor that makes the location of a network entity *relative to a particular message exchange.*

From these definitions it becomes apparent that a network entity can appear to have different locations to other entities. A network entity can also appear to have different locations to the same network entity during different exchanges. For a single exchange though, a network entity has only one location.

> *The **location** of a network entity:*
>
> 1. *Can be determined when this entity is the sink in a message exchange that originated by another network entity.*
> 2. *Is relative to the source entity in this message exchange.*
> 3. *Can be different for different source network entities.*
> 4. *Can be different for the same source network entity between different message exchanges.*

We consider that the only use of the concept of location of a network entity is to be able to contact it. Since a network entity can be contacted in different ways by other network entities, its location has to be relative to these other network entities. Also, since a network entity can be contacted in different ways for different exchanges from the same entity, its location only has a meaning during a message exchange. For this reason, we call the location of a network entity a *relative location*, or just *location* for short.

In many systems, the geographical location of a mobile entity is used in order to determine how the mobile entity can be reached [LM96, LM98]. For example, the geographical location of an employee at a work site (the room that the employee is currently in) can determine which telephone equipment should be used to reach them (the telephone that is closest to the employee in the particular room). In our framework, we would consider the telephone equipment that is eventually used to reach the employee to be the location of the employee. Therefore, in this case, the geographical location of the employee does not map to our concept of location; it is seen as a parameter that allows determining the location of a mobile entity.

3.8 Fixed network entity (FE)

In a communications network a network entity is *fixed* if its location for each of the other network entities does not change between message exchanges. If a network entity can change relative location even for one other entity, then it cannot be considered fixed.

*A network entity is **fixed** if its location, as perceived by each of the other network entities in a communications network, does not change between message exchanges.*

For example, let us assume a communications network that comprises a person *Alpha* who uses a fixed telephone *Terminal1* and a person *Beta* who uses a fixed telephone at home *Home* and another telephone at work *Work*. *Beta* also uses a mobile telephone *Mobile*. Another person *Gamma* is attached to telephone *Terminal2*.

We have eight network entities as shown in Figure 3-5: *Alpha*, *Beta*, *Gamma*, *Terminal1*, *Terminal2*, *Mobile*, *Home* and *Work*. *Alpha* is connected to *Terminal1* via a medium, which is the telephone terminal interface. *Beta* is similarly connected to either *Home* or *Work* and always to *Mobile*. *Gamma* is connected to *Terminal2* in a similar way. *Terminal1*, *Terminal2*, *Home*, *Work* and *Mobile* are all interconnected via the media of the Telephone Company.

We make the following assumptions:

A1. *Alpha* can contact *Beta* either at *Home* or at *Work*.
A2. *Gamma* can always contact *Beta* on *Mobile*, only.
A3. *Gamma* never contacts *Alpha*.
A4. *Alpha* never contacts *Gamma*.
A5. *Terminal1*, *Terminal2*, *Mobile*, *Home* and *Work* are always available and the network of the Telephone Company is always working.

In this example, the location of *Alpha* as perceived by *Beta* is *Terminal1*. The location of *Alpha* does not change for *Beta*. The location of *Alpha* as perceived by *Home* or *Work* is always the same and it is described by *Terminal1*. The location of *Alpha* as perceived by *Terminal1* is always the same, and it is described by the interface that *Terminal1* uses to alert *Alpha* (the ringing of the telephone). Since the location of *Alpha* is the same for every entity of the communications network that contacts *Alpha*, *Alpha* is a fixed network entity.

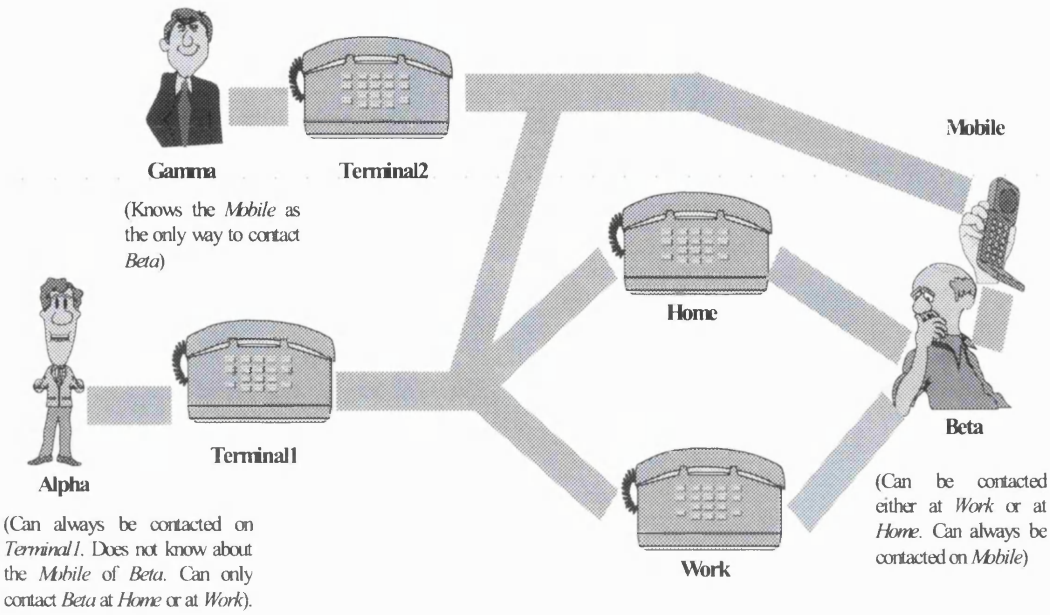


Figure 3-5: Example of Fixed and Mobile network entities.

There are communication networks where if we consider the frequency of reconfiguration of network entities and media it is had to find network entities that are fixed. For example, the communications network of Figure 3-1(c) is reasonable that it may undergo frequent reconfiguration that will make most network entities mobile.

3.9 Mobile network entity (ME)

In a communications network, a network entity that is not fixed is a *mobile* network entity. In other words, a network entity that changes location for any of the other network entities of a communications network between exchanges is mobile.

> *A network entity that is not fixed, is a **mobile** network entity.*

In the example of Figure 3-5 the location of *Beta*, as it is perceived by *Alpha* can be either *Home* for an exchange that takes place when *Beta* is at *Home*, or *Work* for an exchange that takes place when *Beta* is at *Work*. Since the location of *Beta* changes between exchanges for at least one other entity (*Alpha*), *Beta* is not a fixed entity. Therefore, *Beta* is a mobile entity.

3.9.1 Location space

The set of the locations that a network entity may have in relation to another network entity is the *relative location space* or simply *location space* of this entity. The location space of a fixed entity is a set with only one member. The location space of a ME has more than one member for at least one other network entity.

The relative location space of a network entity (N) as perceived by another network entity (A) is denoted by:

$$LS_A(N)$$

In the example of Figure 3-5, the location space of the mobile entity *Beta* as perceived by *Alpha* is a set with two members: *Home* and *Work*. The location space of *Beta*, as it is perceived by *Gamma*, is a one-member set that contains *Mobile*. The location space of *Beta* as perceived by *Mobile* is the medium (the interface) that *Mobile* uses to alert *Beta* when there is a call (the ringing of the telephone).

3.9.1.1 Absolute location space

The union of all the relative location spaces that a network entity has in relation to all other network entities in the network, is the absolute location space of that network entity.

The location space of a network entity (N) is denoted by:

$$ALS(N)$$

In the example of Figure 3-5, the absolute location space of the mobile entity *Beta* is the union of the location spaces of the entities that can contact *Beta*. Therefore, the absolute location space of *Beta* includes: *Work*, *Home*, *Mobile* and the media that *Work*, *Home* and *Mobile* use to alert *Beta* when a call arrives.

3.10 A definition of mobility

Having discussed the concepts of communications networks, fixed network entities, mobile network entities, media and network entity location, we can now attempt a definition of mobility.

Mobility is a quality that allows a network entity to change location in a communications network and, at the same time, to be able to participate in exchanges with other network entities of the same communications network.

3.10.1 Relative speed of a mobile entity

We call *relative speed* or *speed* of a mobile entity the number of times that a mobile entity changes location in relation to another network entity during a period of time, divided by this time.

The average speed of a mobile entity is defined as the measured or anticipated speed of a mobile entity in the long term. The average speed for a mobile entity (M) is denoted by:

$$S_a(M)$$

We call peak speed of a mobile entity the peak speed that this entity can reach, and which can be accommodated by the communications network. The peak speed for mobile entity (M) is denoted by:

$$S_p(M)$$

The speed of a fixed network entity is always *zero*. A fixed network entity can also be called a *zero speed entity*.

3.11 Mobile entity classes

In existing mobile system models mobile entities are grouped according to certain criteria. The grouping of mobile entities can be perceived as made according to their physical characteristics. For example, the grouping of *mobile terminals* refers to a group of entities with the physical characteristics of a terminal that can change point of attachment to the network (see mobile nodes in the Mobile IP architecture, chapter two). Similarly, there can be a group of *mobile users* who have the characteristics of persons who are able to change terminals.

Grouping mobile entities allows to associate certain common properties with them and to treat them in the same way. This assists in the design of mobile systems. For example, in the GSM architecture members of the group *the mobile stations* are identified and distinguished by means of unique identifiers (the IMSI numbers), that comply with a certain format [GSM03.03].

Grouping mobile entities according the their physical characteristics cannot be acceptable on this framework. This is because if physical characteristics were employed for grouping mobile entities, this framework would not be generic. In this framework, the grouping of mobile entities is attempted according to their sharing the same absolute location space. For example, a number of mobile entities like mobile users who share the same absolute location space (e.g. a group of terminals) could be considered a group. This is not the only grouping we can make. However, as the following paragraphs show, this grouping allows us to express certain requirements for the scope of unique identifiers of mobile entities and to outline uniform solutions for locating/contacting entities that share the same absolute location space.

In this section we present different groups of network entities according to the absolute location space that they have. These groups are used when investigating the requirements for mobility in the FoT from the information and computational viewpoints in chapter four.

3.11.1 Mobile entity class (MEC)

We call a mobile entity class (MEC) a group of mobile entities that share an identical absolute location space.

> *A **Mobile Entity Class** is a complete set of the mobile network entities that share an identical absolute location space.*

Or:

$$\forall m_1, m_2 \in MEC, ALS(m_1) = ALS(m_2)$$

Let us consider for example a college that has a number of workstations in the laboratory for the undergraduate students. Every undergraduate student is allowed to use any workstation in the laboratory. If any of the postgraduates or undergraduates at the college needs to contact an undergraduate, they will have to use the Internet *talk* application, and they will be able to contact them on one of these workstations. The undergraduate students are not allowed to use any other workstation in the college.

In this case, the absolute location space for each undergraduate, is the set of the workstations in the laboratory. Since all the undergraduates share the same absolute location space, they consist a Mobile Entity Class.

3.11.2 Network entity class (NEC)

Every MEC can have a corresponding network entity class. A network entity class (NEC) is a group of network entities for which any member of a mobile entity class has the same relative location at a given point in time. It could be the case that a NEC includes members of the MEC too.

> *A **Network Entity Class** is the complete set of the network entities for which the location of any member of a Mobile Entity Class is the same at any given time.*

Or:

$$\forall m \in MEC_1,\ \forall n_1, n_2 \in NEC_1: location_{n1}(m) = location_{n2}(m)$$

In the example of the previous section, we have a MEC that comprises the set of the undergraduate students. As already mentioned, both postgraduate and undergraduate students can contact the undergraduates. In this example, the location of an undergraduate student at any time will appear to be the same to any caller, and it will be the workstation that they are using at that time. Therefore, any undergraduate student will appear to have the same location to all the undergraduate and postgraduate students. In this case, the NEC that corresponds to the MEC of the undergraduates, is the union of the postgraduate and undergraduate students. In this example, the MEC is a subset of the NEC.

3.11.3 Complete Entity Class (CEC)

All the members of a Mobile Entity Class (MEC) share the same Absolute Location Space (ALS). We saw in the previous section that every MEC has a corresponding Network Entity Class (NEC). Here, we define that a MEC also has a corresponding *Complete Entity Class (CEC)*. The CEC is the union of the corresponding NEC and the network entities of the ALS of the MEC. Therefore, the CEC of a mobile entity class includes all the network entities to which the location of any member of the MEC is the same at any given time, plus all the network entities of the absolute location space of the MEC. N.B. a CEC does not include the media that may be members of the ALS.

*A **Complete Entity Class** is the union of the Network Entity Class that corresponds to a Mobile Entity Class and the network entities of the Absolute Location Space of that Mobile Entity Class.*

Or:

$$\forall MEC_1\ \exists (ALS_1, NEC_1, CEC_1),\ CEC_1 = ALS_1 \cup NEC_1$$

In the example of the previous section 3.11.2, the CEC is the union of all the undergraduate students, the postgraduate students and the workstations that the undergraduate students can use.

Let us now assume a mobile entity (M) and another network entity (A) that is having a message exchange with (M). Assume that (M) is a remote entity to (A). For this message exchange, a communications path (CP_1) will have to be created between (A) and (M). Assume that (M) is a member of a mobile entity class (MEC_1), and (A) is a member of the corresponding network entity class (NEC1). If we assume that (A) perceives the location of (M) as (L), then all the other network entities in (CP_1) should perceive (M) at the same location. That is because, for that particular message exchange, the communications path (CP_1) that was created determined (L) as the location of (M). Only the last entity in the CP from (A) to (M), which by definition is (L), perceives the location of (M) differently, and it identifies it by the medium, that connects (M) to (L). Therefore, all the entities of (CP_1), except for (L), perceive (M) to be at the same location (L). And therefore, all the entities of (CP_1), except for (L), belong to the same NEC with (A), that is (NEC_1). We can therefore conclude that:

All the network entities in a CP from a network entity (A) to a mobile entity (M), except for the last one, belong to the same NEC with (A).

Now, the entity (L) of our example is part of the Absolute Location Space of the MEC of (M), MEC_1. From the definition of a CEC, (L) is a member of the CEC of (A), CEC_1. By definition, NEC_1 is also a subset of CEC_1. Therefore:

All the network entities in a CP from a network entity (A) to a mobile entity (M) belong to the same CEC with (A).

3.12 Entity identifiers (ID)

A network entity may need to be identifiable to other network entities. An entity can be identifiable by means of an identifier (ID) which can be used by the other network entities to identify it. A network entity may be identifiable to different entities by different IDs. A single ID must be used to identify a network entity to another network entity.

If we assume a network entity (N) that needs to be identifiable to another network entity (A), (A) can use an identifier to identify (N). This is denoted by:

$$ID_A(N) = \langle \textit{identifier that A uses to identify N} \rangle$$

Another network entity (B) may use a different (or the same) identifier to identify the same network entity (N).

3.12.1 Fixed entity identification (FID)

In this section we investigate the requirements for unique identification of fixed network entities. We examine the necessity for unique identifiers by following a parsimonious approach: we look only at when unique identifiers are absolutely necessary in order to have a message exchange with a fixed network entity.

In order for a message exchange to take place between two network entities, the entities involved may, or may not need to be identifiable. This depends on the nature of the message exchange (one-way, two-way, n-way) and on the media used (unicast or broadcast). If we assume two network entities that are local to each other, we can distinguish between different cases.

E1. *One-way exchange over a unicast medium.* In this case *the sink entity must be identifiable*. This is because, by definition, a unicast medium is able to convey information to a *specified* network entity that is attached to it. The source entity does not need to be identifiable. For example, consider the case of a user (sender) who sends an e-mail message to another user (receiver). In order for this e-mail message to be delivered the receiver of the message has to be identifiable; otherwise the message cannot be delivered to them. On the other hand, the sender does not necessarily need to be identifiable for the e-mail message to be delivered. In this example, the sink entity is identified by means of an e-mail address. Another everyday example is that of post. In order for a letter to be delivered to a person, only the receiver's postal address has to be identified on the envelope; the sender does not necessarily have to be identified. Another example is that of a client of a printing service. In order to send a document to the print server, the address of the print server has to be identifiable; the sender of the document does not necessarily have to be identifiable.

E2. *One-way exchange over a broadcast medium.* In this case the *sink entity does not necessarily need to be identifiable.* By definition, a broadcast medium is able to convey information to *any* network entity attached to it; therefore, the sink entity does not need to be identified. The source entity does not need to be identifiable, either. A good example is the radio network that taxi drivers use in some cities. When there is a request for a taxi at a specific place, the source broadcasts the request to all the taxi drivers, expecting that one of them will pick the task. The driver that eventually picks the task (in our case the sink entity) does not need to be identifiable to the source entity. Also, the source of the request does not need to be identifiable since the content of the request is what matters to the drivers. We assume that only reliable sources can use the radio network. Another example is that of a timeserver that broadcasts time synchronisation messages to a network at specified intervals. In this case, the receivers of the time synchronisation messages do not need to be identified. The sender of the messages (the timeserver) does not necessarily need to be identified either.

E3. *Two-way exchange over a unicast medium.* In this case *the sink entity must be identifiable.* The source entity must be identifiable too. For example, when a user accesses a web server on the Internet, this web server must be identifiable. In order for the information of the web server to appear on the user's terminal, this terminal must also be identifiable. The use of a unicast medium implies that message exchanges can address specified entities only. Another example is the use of a printing service by a client who requires confirmation whether a document it sends to the printing service is printed. In order for this exchange to take place the print server has to be identifiable so that the document is delivered to it. Also, the client (the sender of the document) has to be identifiable too so that the print server can send a confirmation message back.

E4. *Two-way exchange over a broadcast medium.* In this case *neither the source entity, nor the sink entity needs to be identifiable.* When neither the source nor the sink is identifiable, the exchange itself must have an identity, so that the second part of the exchange can relate to the first part. In our example with the taxi drivers, when a driver accepts a request for a taxi, the driver can broadcast on the radio link that this request will be served. The driver does not need to identify the source of the request, since the broadcast medium will convey the reply in any case. The only thing that the driver needs to be able to specify, is which of the requests his reply relates to. Another example is that of a network with many timeservers that broadcast time synchronisation messages at specified intervals. Assume that these timeservers also provide a trigger to clients: when a client uses this trigger the current time is broadcasted by the timeserver out of interval. A time client may broadcast a trigger message to this network asking what the time is. The receivers of the message (the timeservers) do not need to be identifiable. When one (or the first) of the timeservers responds to this message by broadcasting a time synchronisation message, the client that originally triggered it does not necessarily have to be identifiable either.

In this analysis we only examined when it is absolutely necessary for an entity to be identifiable in order for a message exchange to take place. We took the simplest case of two network entities interconnected by one medium. The results of this analysis are summarised in Table 3-1. We see that when a unicast medium connects two fixed entities the source entity *must* be identifiable in a one-way exchange and the sink entity *must also* be identifiable in a two-way exchange.

We will now examine the case where a CP between two network entities includes exactly one other entity. We consider one-way exchanges. In this case the source and the sink entities are remote to each other. As defined before, the location of the sink entity (as perceived by the source entity) is described by the middle entity. There are four possibilities.

	Unicast medium	**Broadcast medium**
One-way exchange	*sink*	none
Two-way exchange	*source/sink*	none

Table 3-1: Identifiable FEs per message exchange per medium.[3]

E1. *Source entity – unicast medium – middle entity – unicast medium – sink entity.* Here, *the sink entity must be identifiable to the source entity.* The middle entity will need the

[3] In the case of a two-way exchange over a broadcast medium, the message exchange itself must be identifiable.

identifier of the sink entity in order to convey the message to it over the unicast medium. The source entity must be able to provide the middle entity with this identifier. Therefore, the sink entity must be identifiable to both the source and the middle entity.

E2. *Source entity – unicast medium – middle entity – broadcast medium – sink entity.* In this case, the middle entity must be identifiable to the source entity so that it can be contacted over the unicast medium. The source entity knows that the location of the sink entity is described by the middle entity. Therefore, *the sink entity must be identifiable to the source entity* so that the association between the sink entity and its location (the middle entity) is valid. The sink entity does not need to be identifiable to the middle entity in the case where the middle entity can contact only entities attached to its broadcast medium. Otherwise, the sink entity must be identifiable to the middle entity, so that the middle entity can create an association between the sink entity and the medium that it has to chose to contact it.

E3. *Source entity – broadcast medium – middle entity – unicast medium – sink entity.* Here, the middle entity needs the identity of the sink entity in order to convey the message to it over the unicast medium. Therefore the source entity must know the identity of the sink entity so that it can pass it to the middle entity. *The sink entity must be identifiable to both the source and the middle entity.*

E4. *Source entity – broadcast medium – middle entity – broadcast medium – sink entity.* This is not a valid case. If both media were broadcast, we could consider a single broadcast medium from the source entity to the sink entity. That is already considered in the analysis with a one-medium-only CP.

If we performed a similar analysis for two-way exchanges, we would conclude that the source should also be identifiable in the first three cases. In the case of E2, the sink entity should also be identifiable to the middle entity.

These are the first two steps of our analysis. The first step involved a CP with only one medium, and the second step involved the next size of a CP with one network entity and two media. These progressive steps of our approach are illustrated in Figure 3-6. From our analysis so far, it becomes apparent, that if we performed more steps by increasing the length of the CPs progressively at each step, we would come to similar conclusions.

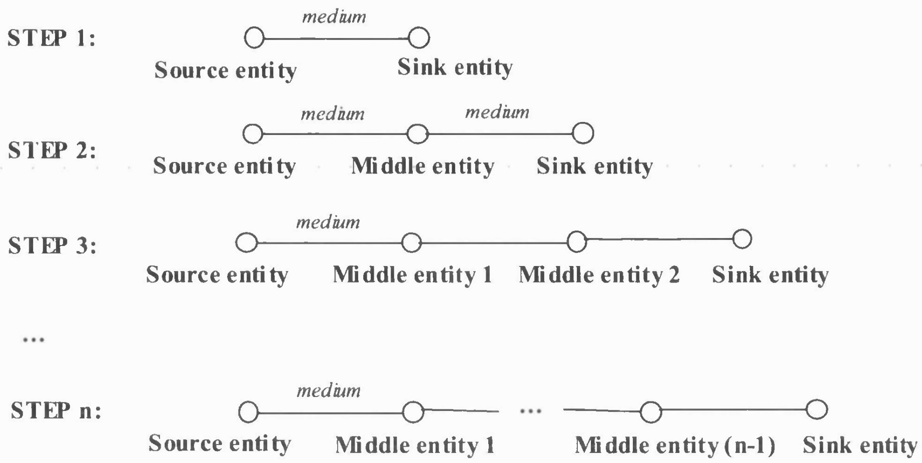


Figure 3-6: Steps 1 ... n of the analysis for fixed network entity identification requirements.

In this section we investigated the requirements on the identities of fixed network entities in the FoT. We can see from this analysis the scope in which fixed network entities have to be identifiable during a message exchange.

We can summarise our requirements on fixed network identification as follows:

Requirements for fixed network entity identification:

R1. *When the communications path between two fixed entities includes at least one unicast medium, the sink entity must be identifiable to the source entity. For two-way exchanges, the source entity must be identifiable to the sink entity too.*

R2. *When the communications path between two fixed entities does not terminate with a broadcast medium, the sink entity must be identifiable to all the network entities of the communications path.*

R3. *When the communications path between two fixed entities terminates with a broadcast medium, the sink entity must be identifiable to all the network entities of the communications path. The only exception is that the sink entity will not have to be identified to the last entity in the communications path if this entity is attached to more than one media.*

From the analysis on our framework so far we can also conclude that media may have to be identifiable to the entities that are attached to them. The identification of these media can be determined by each of the network entities that are attached to them. It is important that if a network entity is attached to *n* different media, then the identifiers that the entity uses for each of these media must be unique in the set of these *n* media. This is to allow a network entity to distinguish between the different media when participating in a message exchange. If a network entity is attached to only one medium, then there is no requirement for identifying it.

> *A network entity that is attached to n different media must use identifiers for them that are unique in the set of n media. This rule applies if n>1.*

3.13 Relating the FoT to existing models for mobility

Having given this specification of the Framework of Terms (FoT) we now relate this work to existing abstract models and mathematical models for mobility. These include the *π-calculus*, the *ambient calculus* and *mobile UNITY*. Background information on these is given in chapter two.

We consider this section important since it outlines how of the FoT terms can relate to concepts found in abstract models for mobility. Therefore, this section can provide a basis for formal work on the FoT. However, formal work on the FoT is in a direction different than the goals of the work in this thesis. The goal of this thesis was to provide a common framework with a generic and technology independent terminology for mobile systems and to describe the requirements for mobility on this framework. The proposed FoT is not a formal model; it is not even a model.

However, it was considered that expressing the definitions of the FoT in formal terms could reduce the ambiguity of its definitions and make them more precise.

3.13.1 Relating the FoT to the π-calculus

An overview of the π-calculus and CCS (Calculus for Communicating Systems) is given in section 2.4.4.1. Here, we relate the FoT to the π-calculus by providing the relationship between FoT concepts and π-calculus ones. Before we proceed it should be mentioned that the objective of the π-calculus work is to provide a mathematical model for distributed concurrent systems. Using the laws of the π-calculus it is possible to check for example whether a design is equivalent to a specification, and whether two systems or agents are equivalent. The objective of the FoT is to provide a basis for describing mobile systems and to make it possible to check if certain mobility requirements are respected in the design of a system. Also, it provides the basis for exchange of ideas and experiences on mobile system design.

A first similarity between the FoT and the CCS (Calculus of Communicating Systems) [Milner89], on which the π-calculus [MPW92] is based, is that the parts of a communication system, the agents, can be expressed at different levels of abstraction. According to Robin Milner [Milner89] an agent can be expressed directly in terms of its interactions with the environment, or indirectly, in terms of its composition from smaller agents. This is quite similar to the FoT; the concept of a *communications network* in the FoT relates to the concept of a *system* in the π-calculus (and CCS). There is a difference however, in that a communications network in the FoT may not have any interactions with the environment, while a CCS system can. If any other entity interacts with a communications network in the FoT, then that entity should be part of the same communications network.

The concept of a *network entity* in the FoT relates to an *agent* in the π-calculus (CCS). Both FoT network entities and CCS agents can be senders or receivers of messages. However, the concept of a *medium* in the FoT relates to an *agent* in the π-calculus (CCS) and it cannot relate to a π-calculus handshake. The reason for this is the same reason that channels cannot be distinguished from agents according to CCS [Milner89]. Namely, the FoT media can have an active role in terms of providing broadcast or unicast functionality, and therefore, cannot be related to CCS *handshakes*. However, the FoT media cannot be a Source or a Sink of messages.

The FoT concept of *communications network topology* is similar to the concept of agents possessing *links* to other agents in the π-calculus. The difference is that a FoT topology graph, like the ones of Figure 3-3, cannot contain open links (media), while a π-calculus system represented in a flow graph [Milner89] can. The concept of a *dynamic* communications network topology of the FoT can be supported in a similar way in the π-calculus by means of passing link names between agents.

Apart form these relationships there are some general differences between the FoT and the π-calculus concepts. One of them is that names are used in the π-calculus for links or ports to agents. In the FoT, names are used for identifying network entities and media. This is because the FoT intended to express routing and addressing requirements, where a message can be addressed to a network entity (a sink of a message exchange) and this message can be routed to this destination accordingly. Therefore, the destination of a message exchange is identified in order to support routing and addressing.

Another difference is that the π-calculus extended the CCS in order to allow passing of link names between agents. This can allow for a dynamic configuration of links in a system. The π-calculus can also express transitions that describe *scope extrusion* and *scope intrusion* for names of links [MPW92] in a system. On the other hand, the scope of the FoT is wider. The FoT intends to allow not only for dynamic topology configuration but for *addressing and routing* in a new topology, for guarantying a *scope of uniqueness* for names of network entities and media in dynamic topologies, and for *describing functional solutions* for these.

A concept of *location* has been defined for *flow graphs* or *interaction diagrams*, (the graphic counterpart of terms in the π-calculus) by Joachim Parrow [Parrow95]. According to this definition, a location relates to the access points and links to agents. There are *free locations*, which represent access points or links that are known by means of a unique name. Locations can provide input and output to each other. This concept of location is different from the concept of location in the FoT. In the FoT, the location is not associated with a link or access point. It is rather associated with the mediating entities when an interaction between two network entities takes place. For this reason, a FoT location is identified by either the *last entity* encountered in a message exchange between a source entity and a sink entity (where the sink is remote to the source) or by the *medium* between two entities (where the sink is local to the source). Also, the location of a sink entity for a message exchange is *relative* to the source, and it may change from one exchange to another.

Finally, the difference in scope between the π-calculus and the FoT is also evident by issues that standard π-calculi do not address, which Peter Sewell points out [Sewell99]. These issues include point-to-point and multicast communication, code or agent migration and quality of service. These issues can be addressed in the FoT by using existing material in this thesis or by building on it. Also, different aspects of communication systems can be captured in the FoT by using the different ODP viewpoints. However, the FoT does not provide a mathematical model for investigating these issues while π-calculi do.

3.13.2 Relating the FoT to Mobile Ambients

The FoT and the ambient calculus address the issue of mobility in a different scope. The ambient calculus aims to extend the π-calculus with the concept of administrative domains and security. The concept of an ambient (which can represent an administrative domain) within which processes are confined serves this purpose. Other ambients can move in or out of an ambient subject to a set of capabilities that the latter can give out to them. In this way the crossing of administrative domains can be controlled, as explained in section 2.4.4.2.

On the other hand, the FoT addresses mobility in a different scope that intends to include administrative domains and security next to a number of other concepts, which include communication primitives, identification of mobile entities, routing, addressing and location mechanisms for mobile entities. In this sense, the concepts of the mobile ambients can be seen as potentially complimentary to the FoT work. The ideas of the mobile ambients could be introduced in the FoT too.

Since the ambient calculus is an evolution of the π-calculus, the relationship of the FoT to the π-calculus (see section 3.13.1) can be translated into the ambient calculus. Since a channel of the π-calculus can relate to a mobile ambient [CG98], the FoT concept of a *medium* can relate to a *mobile ambient.* In a similar way, a *network entity* of the FoT could relate to a *mobile ambient* of the ambient calculus, since a FoT network entity may be able to impose access control to its resources.

The idea of using mobile ambient hierarchies can apply to the FoT. Administrative domains (which are not yet part of the FoT) can include a number of network entities and media or other administrative domains (*administrative domain hierarchy*). This concept of administrative domains relates to mobile ambients that contain other sub-ambients (*ambient hierarchy*).

If in the ambient calculus the concept of a *location of a process (agent) or location of an ambient* is that of another ambient that contains them (as it seems to be implied from the distinction between local and remote communication primitives [CG98], then this concept does not relate to the FoT concept of a location. However, if the administrative domains are introduced in the FoT it may be the case that the FoT concept of location is enhanced with administrative domains serving as locations.

The idea that the *name* of an ambient in the ambient calculus can give access to the ambient itself is very different from the FoT concept of an identity. The FoT identity can be globally known; access control to an identified entity is a functional requirement.

Finally, the concepts of *entry*, *exit* and *open capabilities* cannot relate to FoT concepts at the moment, since administrative domains and security are not yet part the FoT. However, the concept of capabilities could serve as a useful basis for a potential FoT extension.

3.13.3 Relating the FoT to Mobile UNITY

Mobile UNITY provides the means for expressing the interactions among mobile components (programs) of a system and subject them to a rigorous formal verification against certain requirements. Failure and disconnection of mobile components can be successfully modelled in mobile UNITY.

Mobile UNITY has been successfully used for the verification of Mobile IP [MR99] and for expressing code mobility [PRM97].

We consider that the FoT could similarly be expressed in Mobile UNITY. However, the FoT by itself is not a prescriptive architecture; it is rather a framework that provides a generic terminology for mobile systems and a number of requirements that are descriptive of mobile systems. Therefore, it is really the solutions that are described in FoT terms (mainly the computational viewpoint solutions) that could be expressed in mobile UNITY and rigorously verified. This could provide useful further work.

3.14 Expressing the FoT in π-calculus terms

The Framework of Terms (FoT) is described in plain English and its definitions are accompanied by a number of examples. Due to this, some definitions may be ambiguous. In this section, in order to reduce the ambiguity of the framework we express some key concepts of the FoT using π-calculus notation. The attempt is not to map the FoT to the π-calculus but to express its terminology in π-calculus terms. The definitions of this section follow the work in section 3.13.1 that relates the FoT to the π-calculus. A brief tour of the π-calculus notation is provided in Appendix B.

3.14.1 Communications Network

The FoT communications network can be expressed as a system of inter-linked agents in the π-calculus. There are no links from the environment to the system or from the system to the environment.

3.14.2 Network Entity

A FoT network entity can be expressed by an agent in π-calculus terms. It can possess a number of links to/from other agents that express media (see section 3.13.1). A π-calculus agent that expresses a network entity cannot be linked to another agent that expresses a network entity directly. A π-calculus agent that expresses a network entity in this section is also called a *network entity agent.*

3.14.3 Media

A FoT medium can be expressed by an agent in π-calculus terms. It can possess a number of links to/from other network entity agents. A π-calculus agent that expresses a medium cannot be linked to another entity that expresses a medium directly. A π-calculus agent that expresses a medium in this section is also called a *medium agent.*

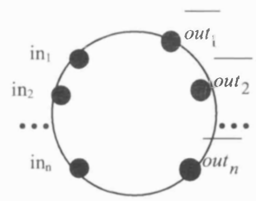


Figure 3-7: A FoT medium as a π-calculus agent.

Formulas (1) and (2) describe the behaviour of unicast and broadcast media respectively. Figure 3-7 illustrates a medium agent. We assume that the medium is attached to (n) other network entity agents for which it has input and output links for communication with them.

$$(1)\ \mathrm{Uncst}(ne, msg) \overset{def}{=} \overline{out_{ne}}.\mathrm{Uncst}(ne, msg)$$

$$(2)\ \mathrm{Brdcst}(msg) \overset{def}{=} (\overline{out_1}\langle msg\rangle \mid \overline{out_2}\langle msg\rangle \mid \ldots \mid \overline{out_n}\langle msg\rangle).\mathrm{Brdcst}(msg)$$

3.14.4 Communications Network Topology

A FoT communications network topology can be represented by a flow graph in π-calculus terms. A π-calculus flow graph will show the agents of a system and the configuration of the linkage between them.

3.14.5 Message exchange

A FoT message exchange can be expressed by the passing of names between agents in π-calculus terms via the links to each other that they posses. In Figure 3-8 a message exchange between a source network entity agent and a sink network entity agent is illustrated. The message is passed from the source_out link of the Source agent via a number of agents to a sink_in link of the Sink agent. Formulas (3) and (4) express this message exchange. N.B. the agents between the Source and the Sink entity are media agents and network entity agents. N_1, N_3, … N_n are medium agents, while N_2, N_4, … N_{n-1} are network entity agents.

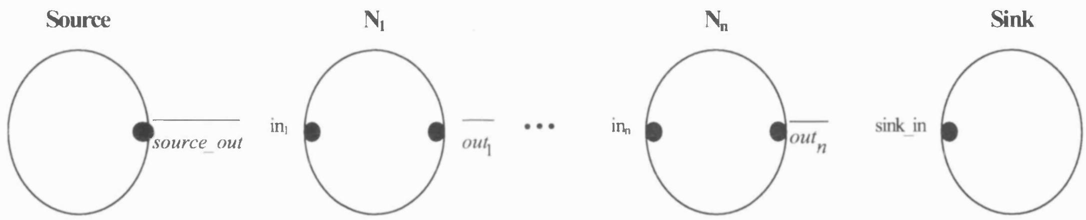


Figure 3-8: A FoT message exchange as an exchange among π-calculus agents.

$$(3)\ \mathrm{Exchange}(msg) \overset{def}{=} \overline{source\_out}\langle msg\rangle.\mathrm{Exchange}'(msg).sink\_in(msg)$$

$$(4)\ \mathrm{Exchange}'(msg) \overset{def}{=} in_1(msg).\overline{out_1}\langle msg\rangle.in_2(msg).\overline{out_2}\langle msg\rangle \ldots in_n(msg).\overline{out_n}\langle msg\rangle$$
$$n = 1,3,5,\ldots$$

3.14.6 Communication path

When a message exchange takes place between a Source network entity agent and a Sink network entity agent the configuration of the agents that are involved in the message exchange and which lie between the Source and the Sink form a communication path.

For the message exchange of Figure 3-8 the communication path (CP) is the sequence of the linked agents: N_1, N_2, … N_n.

3.14.7 *Location*

For a particular message exchange, the location of a Sink network entity agent is the last agent encountered in the communication path from the Source network entity agent to the Sink network entity agent. This is shown in Figure 3-9. Formulas (5) and (6) express a message exchange including the location.

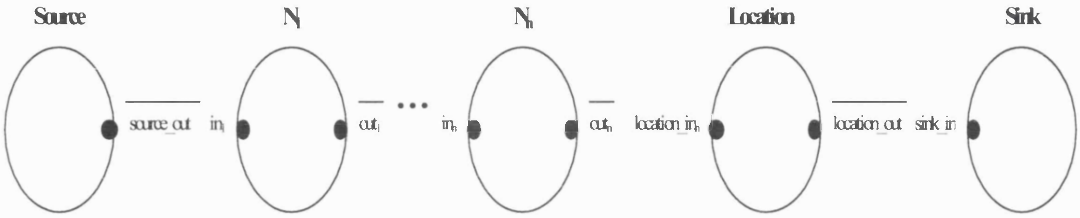


Figure 3-9: The FoT concept of location in π-calculus terms.

$$(5)\quad \text{Exchange}(msg) \overset{def}{=} \overline{source\_out}\langle msg\rangle.\text{Exchange}''(msg).location\_in(msg).\overline{location\_out}\langle msg\rangle.sink\_in(msg)$$

$$(6)\quad \text{Exchange}''(msg) \overset{def}{=} in_1(msg).out_1\langle msg\rangle.in_2(msg).out_2\langle msg\rangle...in_n(msg).out_n\langle msg\rangle$$
$$n = 0,2,4,...$$

In formulas (5) and (6) if n=0, the Sink network entity agent is *local* to the Source network entity agent. If n>0, then the Sink network entity agent is *remote* to the Source network entity agent.

3.14.8 *Fixed network entity*

A FoT fixed network entity can be expressed by a network entity agent in π-calculus terms, for which any message exchange in the system from a Source network entity agent to it can be expressed by formula (7) at any time.

$$\forall source\_out,$$

$$(7)\ \text{Exchange}_f(source\_out, msg) \stackrel{def}{=} \overline{source\_out}\langle msg\rangle.\text{Exchange}''(msg).location\_in(msg).\overline{location\_out}\langle msg\rangle.sink\_in(msg)$$

3.14.9 Mobile entity

A FoT mobile network entity can be expressed by a network entity agent in π-calculus terms, for which the message exchanges in the system from different source network entity agents to it can be expressed by formula (8). In this section we call an agent that expresses a mobile entity *mobile entity agent.*

$$\forall source\_out,$$

$$(8)\ \text{Exchange}_m(source\_out, msg) \stackrel{def}{=} \overline{source\_out}\langle msg\rangle.\text{Exchange}''(msg).(\sum_{i\in\{1,\ldots,l\}} location\_in_i(msg)\overline{location\_out_i}\langle msg\rangle).sink\_in(msg)$$

$$l > 1$$

3.14.9.1 Absolute location space

The absolute location space of a mobile entity agent is the set of entities that can be the locations of a mobile entity agent for any message exchange from any Source network entity agent to it. In formula (8), the absolute location space consists of the (l) number of agents the links of which appear in the sum.

3.14.10 Mobile entity class

A FoT Mobile Entity Class (MEC) can be expressed in π-calculus terms by the set of mobile entity agents which share the same absolute location space. Message exchanges from any Source network entity agent of the system to any member of a MEC can be expressed by formula (9).

$$\forall source\_out, j \in MEC,$$

$$(9)\ \text{Exchange}_j(source\_out, msg) \stackrel{def}{=} \overline{source\_out}\langle msg\rangle.\text{Exchange}''(msg).(\sum_{i\in\{1,\ldots,l\}} location\_in_i(msg)\overline{location\_out_i}\langle msg\rangle).sink\_in(msg)$$

3.14.11 Network entity class

A FoT Network Entity Class (NEC) that corresponds to a FoT Mobile Entity Class (MEC) can be expressed in π-calculus terms by the set of network entity agents that when they are the source of a message exchange to any mobile entity agent of the corresponding MEC this exchange can always be expressed by formula (10) at any given time.

$$
\begin{aligned}
&\forall source\_out, j \in NEC, Sink \in MEC \\
(10)\quad & \text{Exchange}_j(sink\_in, msg) \stackrel{def}{=} \\
&\overline{source\_out}\langle msg\rangle.\text{Exchange}''(msg).location\_in_i(msg)\overline{location\_out_i}\langle msg\rangle.sink\_in(msg)
\end{aligned}
$$

3.14.12 Complete entity class

A FoT Complete Entity Class (CEC) that corresponds to a FoT Network Entity Class (NEC) can be expressed in π-calculus terms by the union of the agents of Network Entity Class (NEC) and the network entity agents of the absolute location space that mobile entity agents of the corresponding MEC have.

3.15 Summary

In this chapter we presented our Framework of Terms (FoT) in which we define mobile network entities and fixed network entities in a communications network. We investigated concepts of location, mobility and requirements for the identification of fixed network entities in this framework. With examples we showed how this framework can map to a variety of real environments. Examples of mapping the FoT to the Mobile IP architecture, to the GSM architecture and to the role mobility architecture of Prospect are given in chapter five. We also described how the FoT concepts relate to other mathematical and abstract models for mobility, although the FoT is not a model in itself. In particular, we presented how the FoT concepts relate to those of the π-calculus, the mobile ambients calculus and mobile UNITY.

As a consequence of the informal nature of the FoT, there is a certain amount of ambiguity in the FoT definitions. In order to reduce this ambiguity FoT definitions were also expressed in the π-calculus terms.

Although we discussed some requirements regarding the identification of fixed network entities, we intentionally did not discuss any of the requirements for mobile network entities. In the following chapter we examine the requirements that mobility introduces in this framework. We look at mobile entities from two different ODP [ISO-ODP1] viewpoints: the information viewpoint and the computational viewpoint. From the information viewpoint we discuss the requirements concerning the identification of mobile network entities in our framework. From the computational viewpoint we discuss requirements in terms of components to support mobility for network entities. Some engineering viewpoint issues are also discussed.

4. Managing mobility in the Framework of Terms

In the previous chapter we presented a technology independent Framework of Terms (FoT) that defines fixed and mobile entities, and we discussed some of the requirements regarding the identification of fixed network entities using the terms of the FoT. That was the first step of our approach, as it is presented in the introduction of this thesis.

This chapter documents the second step of our approach. Here we investigate the requirements for managing mobile entities using the terminology of the FoT. We examine the requirements for managing mobile entities from two ODP viewpoints: the information viewpoint in section 4.1 and the computational viewpoint in section 4.2. From the information viewpoint we investigate the minimum set of requirements for the identification of mobile network entities: we examine the need for identifiers, the scope of uniqueness of those identifiers and the structure that they should have. From the computational viewpoint we decide a minimum set of computational entities that are required to support mobile entities. We present our results in the form of design patterns using a CORBA design pattern template [MM97]. We also suggest how a combination of these design patterns can fix some of the drawbacks that each design pattern separately has. Some engineering viewpoint issues are addressed when discussing the pros and cons of the suggested design patterns.

4.1 Information viewpoint

Starting with the information viewpoint requirements in this section, we investigate the identity for mobile network entities using the FoT terminology. We discuss the uniqueness of these identities and the scope of their uniqueness. Throughout our analysis we aim for the minimum requirements for mobile entities.

4.1.1 Identity of mobile network entities

From the definition in section 3.9, a mobile network entity is able to change its location in relation to other network entities during the course of time. This adds extra requirements to the identity of a mobile entity. We can perform an analysis similar to the one for fixed network entities of section 3.12.1 to see what these extra requirements are.

Let us assume that we have a network entity (A) that is connected to entity (M) via two separate media. Both media are identifiable. (M) can be connected to either of the two media at a time as shown in Figure 4-1. (A) has to locate (M) in order to have a message exchange with it. The location of (M) can be obtained prior to a message exchange, or while a message exchange is being carried out. There are the following possibilities now:

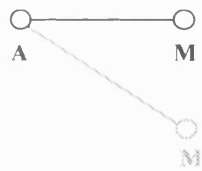


Figure 4-1: CP without network entities, where the sink entity is mobile.

B1. *Both media are unicast.* In this case, (M) must be identifiable. This conclusion can be derived from the analysis of the previous chapter on the identity of fixed network entities.

B2. *One medium is unicast, while the other medium is broadcast.* In this case, (A) must be able to distinguish on which medium (M) is located, each time. In order to make the association between (M) and its location, (A) needs to have an identity for (M). Therefore, (M) must be identifiable to (A). If (A) has the ability to send each message to both media every time, assuming that (M) will be in one of them, then this is a case of two entities linked by a single broadcast medium, which was examined in the previous chapter. In this case, (M) cannot be considered a mobile entity by (A).

B3. *Both media are broadcast.* In this case, (M) must be identifiable, so that the association between (M) and its location is valid for (A). In this way, (A) will be able to decide which of the two broadcast media to use in order to reach (M). It should be noted that if (A) broadcasts to both media in order to reach (M), then:

- If (A) always broadcasts to both media this means that it treats them as a single broadcast medium.
- If (A) broadcasts to both media to reach (M), then according to the definitions of sections 3.8 and 3.9 (M) does not change location in relation to (A) when it goes from one medium to the other and therefore this analysis does not apply to it.

From this analysis we can conclude that a mobile entity must always be identifiable to the other local network entities in order for the latter to be able to contact it.

A mobile entity must always be identifiable to the other network entities to which it is local.

Now, let us assume a CP with three network entities: (A), (L) and (M) as sown in Figure 4-2. (M) is a mobile entity which at the moment has a relative (to entity (A)) location described by (L). Therefore:

$$location_A(M)=L$$

(M) must be identifiable to (A), so that (A) can specify to (L) which entity is to be contacted. (M) must also be identifiable to (L), as derived from the analysis for the local network entities.

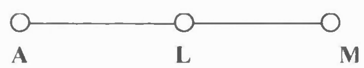


Figure 4-2: CP with one network entity, where the sink entity is mobile.

Let us now assume a CP with four network entities as shown in Figure 4-3. The mobile entity (M) must be identifiable to (A) so that (A) can specify to (B) which entity is to be contacted. (M) must be identifiable to (B) as we concluded from the analysis of the previous paragraph. (M) must be identifiable to (L) as well, as we concluded in the analysis for the local mobile entities at the beginning of this section.

Figure 4-3: CP with two network entities, where the sink entity is mobile.

If we try the same analysis for longer CPs we will come to the similar conclusions. Therefore:

A mobile entity must always be identifiable to all the network entities on a communications path that terminates at this entity.

We can also conclude that a remote mobile network entity must always be identifiable to the last entity on the communications path that terminates at this entity. From the definition of section 3.7.2, this last entity of the communications path serves as the location of a remote mobile entity. Therefore:

A remote mobile entity must be identifiable to the entity that serves as its location.

Network entities can be identifiable by means of *identifiers*. In the following paragraphs we examine the uniqueness of these identifiers.

4.1.2 Unique identifier (UID)

Assume a mobile entity class MEC_1 and its corresponding complete entity class CEC_1. By definition (see section 3.11.1), all the mobile entities of MEC_1 have the same absolute location space, ALS_1. Also by definition (see section 3.11.3), the members of CEC_1, are the union of the network entities of the absolute location space of MEC_1 and the network entities that perceive each mobile entity of MEC_1 at the same location at any given time.

From section 4.1.1, we know that each mobile entity must be identifiable to all the other network entities on a communications path for any message exchange. Also, from section 3.11.3 we know that if a network entity of a CEC has a message exchange with a mobile network entity of the corresponding MEC, the network entities of the communications path, are members of the same CEC, too.

An identifier can be used by the network entities of the CEC to identify each mobile entity of the MEC. This identifier must identify each mobile entity uniquely in its MEC set. *This is essential because the members of a MEC share the same location space and,* *consequently, it is required to be able to distinguish between entities that may share the same location at a given time*. Therefore:

$$\forall n \in CEC_1,\ \forall m_1, m_2 \in MEC_1: ID_n(m_1) \neq ID_n(m_2)$$

We can conclude that:

Each member of a CEC must use unique identifiers for the members of the corresponding MEC.

A question arises on whether the identifiers used by the members of the NEC_1 for the members of the MEC_1 have to be the same. In other words, we have to question whether:

$$\forall n_1, n_2 \in CEC_1,\ \forall m \in MEC_1: ID_{n1}(m) = ID_{n2}(m)$$

Considering the conclusion of the previous section, according to which a mobile entity must be identifiable to all other entities of a communications path to it, and bearing in mind that the network entities on the communications path essentially belong to the same CEC, we can determine that:

The members of a CEC may use different identifiers for the same member of the corresponding MEC. When a communications path is set up between entities that use different identifiers for the same mobile entity a mapping must take place between the IDs that different entities use.

For example, consider a message exchange and a communications path that is set up between a source network entity (A), a middle network entity (B) and the sink mobile entity (M). The message exchange takes place between (A) and (M) via (B). If (A) and (B) use different identifiers for (M), the network entity (B) has to map the $ID_A(M)$ to the $ID_B(M)$. But this would require (B) to be aware of both $ID_A(M)$ and $ID_B(M)$. This would make the handling of the message exchange from (A) to (M) more complex, and it would require that some of the network entities of the CP have knowledge of more than one identifier for (M).

The *simplest* case would require that each of the entities in the (CP) from (A) to (M) use *the same identifier* for (M). From section 3.11.3 we know that all the entities in the (CP) from (A) to (M) belong to the same CEC. If we *generalised* this requirement for all the members of the CEC we would secure that this requirement is always satisfied. Therefore:

> *In the simplest case, all the members of a CEC use the same unique identifiers for the members of the corresponding MEC.*

This case is considered in the rest of our analysis.

4.1.3 Scope of uniqueness

In the previous section, we considered that the mobile entities of a MEC were identified with unique identifiers to the members of the corresponding CEC. The scope of uniqueness for the identification of a mobile entity lies within the MEC to which the mobile entity belongs. In the example of Figure 4-4, the entities of CEC_1 must use identifiers for mobile entities M_1, M_2, M_3 and M_4 of MEC_1 that are unique in MEC_1.

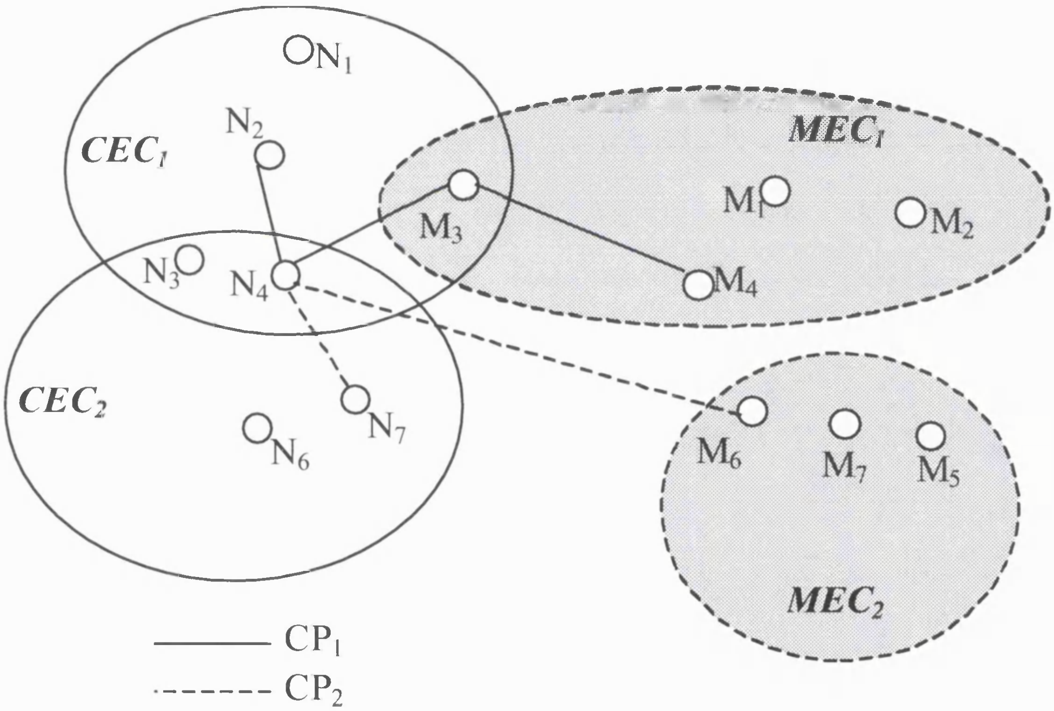


Figure 4-4: CPs between network entities of different NECs and MECs.

It can be the case that a network entity belongs to more than one CEC. In the example of Figure 4-4, we have two MECs and their corresponding CECs. There is MEC_1 with its corresponding CEC_1, and MEC_2 with its corresponding CEC_2. We can see that M_3 of

MEC_1 is also a member of CEC_1. We can also see that CEC_1 and CEC_2 have two entities in common: N_3 and N_4.

In this example we assume two message exchanges. One message exchange, which establishes CP_1, takes place between N_2 of CEC_1 and M_4 of MEC_1. The other message exchange, which establishes CP_2, takes place between N_7 of CEC_2 and M_6 of MEC_2.

For the first message exchange, N_2 must identify to N_4 the entity that is to be contacted, M_4. The entity N_2 is a member of CEC_1 (like N_4 is) and the identifier that N_2 uses for M_4 identifies M_4 uniquely in MEC_1. However, it could be the case that the same identifier is used to identify another entity of MEC_2, uniquely in MEC_2. Since N_4 is a member of both CEC_1 and CEC_2 it should be able to distinguish which MEC the given mobile entity identifier corresponds to. N_4 needs an identifier for M_4 that is unique in both CEC_1 and CEC_2.

Similarly, during the second message exchange, N_4 must receive from N_7 an identifier for M_6 that identifies it uniquely in both CEC_1 and CEC_2 in order to avoid confusion.

This example demonstrates that when a message exchange takes place between a network entity of a CEC and a mobile entity of the corresponding MEC, it is possible, that an entity that is on the communications path that is established, belongs to more than one CEC. In this case, it is imperative to specify not only the mobile entity but also the MEC that it belongs to. Therefore, *a MEC (or CEC) identifier (unique in the communications network) is necessary in order to guarantee a globally unique ID for a mobile entity.* This globally unique ID (GID) could contain two fields: one that identifies a mobile entity uniquely in its MEC and one that identifies the MEC (or CEC) uniquely in the communications network.

GID = <MEC identifier> <mobile entity identifier >

Therefore:

*A **Globally unique Identifier (GID)** identifies a mobile entity uniquely in a communications network.*

Of course, if a CEC of a communications network does not have common members with any other CEC of the communications network the requirement for a GID does not apply.

That is because during any message exchange between a member of the CEC and a member of the corresponding MEC all the entities that will be in the CP will be part of the same CEC only.

The requirement for a Globally unique Identifier (GID) for the members of a MEC does not apply if the corresponding CEC has no common members with any other CEC of the communications network.

Table 4-1 summarises the requirements on unique identifiers for mobile entities in our framework.

Requirements for the identification of mobile entities in the FoT
R1. A mobile network entity must be identifiable to any other network entity in order for the latter to be able to contact it.
R2. The identifier that is used to address a mobile network entity must be unique in the MEC to which this entity belongs.
R3. In the simplest case, where each member of a CEC knows and uses only one identifier for each mobile entity of the corresponding MEC, all the members of the CEC use the same identifiers for each entity of the corresponding MEC.
R4. If no member of a CEC belongs to any other CEC of the communications network, the identifiers used for the mobile entities of the corresponding MEC should be unique in this MEC.
R5. If (R4) is not true, then a Globally unique Identifier (GID) should be used to address a mobile entity. The GID could contain two fields: one field identifies the MEC (or CEC) uniquely in the communications network, while the other field identifies a mobile entity uniquely in the MEC.

Table 4-1: Requirements for the identification of mobile network entities in the FoT.

4.2 Computational viewpoint

The mobility of network entities as defined in the FoT introduces some problems from the computational viewpoint. A mobile entity can change location. While changing location, a mobile entity may be unavailable or unreachable. In order for other network entities to contact a mobile entity the following problem areas must be investigated:

A1. Locating the mobile entity.

A2. Coping with the fact that a mobile entity may be unavailable while on the move.

We do not claim that with these two problem areas we cover all the requirements for mobility. There can be many other aspects such as security considerations. We do claim though, that these two areas are the bare basis for mobility support in any system. We consider these two areas the mean common denominator of the requirements for mobility support.

In this analysis, we investigate the necessary support for these areas, in terms of computational facilities. We outline what ODP computational objects [ISO-ODP1] may be required and what functionality these objects should offer. We facilitate two design patterns to describe the computational level support for this purpose. The two design patterns that we describe are:

DP1. A "Mobile Entity Locator" (MEL) design pattern, in section 4.2.1.

DP2. A "Mobile Entity Agent" (MEA) design pattern, in section 4.2.2.

These two design patterns are not axiomatic as solutions for supporting mobility. We consider axiomatic only the two problem areas that these two design patterns attempt to solve (these are the areas A1 and A2 mentioned above). Therefore, we state as an axiom that these two problem areas must be tackled in order to support mobility.

The design patterns that we provide (DP1 and DP2) are examples of how these two problem areas can be tackled, providing the pros and cons of different solutions. They are *analysis design patterns* in this sense. Based on such an analysis, it is expected that mobile system designers will provide more solutions in a design pattern form that may well support mobility more efficiently than the ones provided.

However, it is not implied that if a design pattern has many pros then it is "better" than another one that has fewer pros. In order to determine that, performance information would be needed and this information can be different from system to system. An engineering viewpoint analysis could give an insight to system performance.

The design patterns that we provide tackle the problem of mobility support using FoT terminology, and therefore, they should map to a diversity of real systems that the FoT terms can map to.

We use a CORBA design pattern template for the specification of the design patterns. This template is the one suggested by Thomas J. Mowbray and Raphael C. Malveau [MM97]. We found this template appropriate for describing our design patterns in FoT terms, since it can accommodate design patterns of a wide scope and applicability.

The design pattern template that we use allows us to give examples of the application of these design patterns to real systems and to provide the pros and cons that each of them may have. We also discuss some engineering viewpoint issues that may arise when implementing these design patterns for real systems.

In section 4.2.3 we provide a combination of the two design patterns to a new design solution. We show how combining these two design patterns can cure some of the cons that each of them separately has. This is to show that specifying solutions in design pattern form and in FoT terms can be useful in analysing the pros and cons of certain design solutions and curing some of the drawbacks of certain designs. However, a design solution that has fewer drawbacks does not necessarily mean that it is "better" than another one with more drawbacks; performance information would be needed to justify such a claim. Such information can vary.

4.2.1 Mobile-Entity-Locator (MEL) Design Pattern

This design pattern tackles the first problem area that mobility introduces in our framework: how mobile entities can be located. Here follows the design pattern specification in the CORBA design pattern template [MM97]. A concise description of this design pattern template is given in Appendix A.

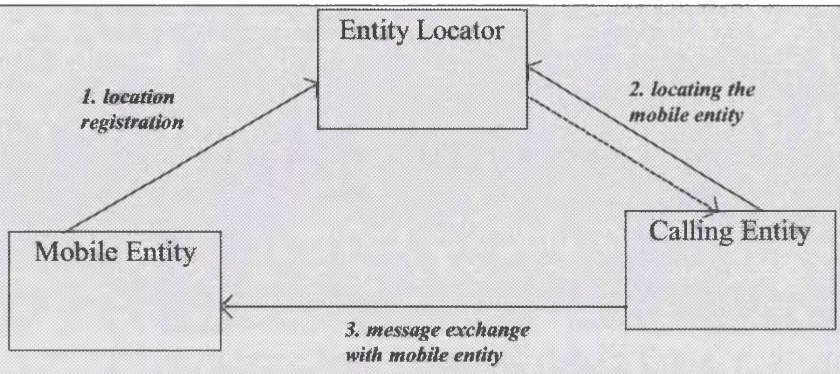
Most Applicable Scale: Enterprise

Solution Type: Technology

Solution Name: Mobile Entity Locator (MEL)

Intent: To outline a generic set of computational objects required in order for mobile entities to locate/be located by other entities in a communications network.

Diagram



Primal Forces: Management of Functionality, Complexity and Change.

Applicability at This Scale

1. A communications network contains a large number of calling entities, each of which may call a large number of mobile entities.

2. Keeping track of the location of a mobile entity in a communications network.

3. A mobile entity updates its location registration only at one place each time it moves.

4. A mobile entity is unavailable while changing location.

Solution Summary

- Provide an entity, the entity locator, which holds the current location of mobile entities. The entity locator is at a location that is well known to all possible calling entities.

- Each mobile entity has to register/update their location with the entity locator every time they move.

- A calling entity can obtain the location of a mobile entity that they wish to contact from the entity locator. The entity locator can enable access control on the calling entities.
- Once a calling entity has the location of a mobile entity, they can have a message exchange with that mobile entity.
- The entity locator can be either a single engineering object [ISO-ODP1] or it can comprise a number of distributed engineering objects that work like a distributed database.
- There can be one engineering object that serves as the entity locator for all the members of a Mobile Entity Class (MEC). All the members of a MEC share the same Absolute Location Space (ALS) and they appear to have the same location to the members of their Network Entity Class (NEC).
- There can be entity locators with the scope of a MEC class. In this case, they could return the location of a mobile entity only to members of the corresponding NEC class.
- There can be entity locators with the scope of the communications network. In this case, they would require a Globally Unique Identifier (GID) to be able to return the location of a given mobile entity. That is because if a mobile entity belongs to more than one MEC, its location may be different for the calling entities depending on which NEC the calling entities belong to.

Benefits

A mobile entity has to update its location record with only one other entity, the entity locator.

A calling entity only has to contact a single party, the entity locator, in order to retrieve a mobile entity's location.

Consequences

There is some processing overhead since a calling entity has to contact the entity locator before they can have a message exchange with a mobile entity.

A mobile entity cannot be contacted if the entity locator is unavailable to the calling entity.

At a given time, a mobile entity may not be able to update its location with the entity locator if the latter is unreachable at that time.

The record of the location of a mobile entity, that the entity locator maintains, will be obsolete during the time that a mobile entity is changing location.

If the entity locator is a distributed database with cached records of entity locations, it may take some time before an entity location update reaches all the cached copies of the location records for that entity. During this time, the cached location records will be obsolete.

When a calling entity obtains the location of a mobile entity from the entity locator, the mobile entity may change location before the calling entity starts the message exchange.

Variations on the Solution

- The entity locator could enable access control and offer confidentiality over its records. However, it may be hard to provide customised access control for retrieving the location of individual mobile entities, due to the scale of this scheme.
- Another variation is that, when a mobile entity is about to change location, it can invalidate its location with the entity locator first. After the mobile entity changes location it has to re-register its location with the entity locator. This would prevent the location record of the mobile entity from being obsolete, while this entity is on the move. On the other hand, this

solution is time-dependent; the mobile can invalidate its location if no-one is hanging onto a stale location.

Rescaling to Other Levels

This pattern can apply to the global architecture level without important changes. At that level, the consequences of cached location records and of location records becoming obsolete just after retrieving them, take more emphasis. Only the assumption that the mobile entities do not change location too frequently can reduce the effects of these consequences on the global level.

On a system or application architecture level, the importance of the entity locator is reduced by the fact that the number of calling or mobile entities may be smaller. The importance of an entity locator depends on the complexity of individual systems or applications.

Related Solutions

- The OMG CORBAService Naming Design Pattern, by T. J. Mowbray and R. C. Malveau. This design pattern allows the association of an object reference with a specific name and it can offer location transparency.

- The Agent Design Pattern, by T. J. Mowbray and R. C. Malveau. This design pattern simplifies client access to disparate information services and provides a uniform, consolidated access to disparate services.

- The SINGLETON design pattern by E. Gamma et al [GHJV97], which ensures that each class only has one instance and provides a global point of access to it. This design pattern can also allow multiple instances, if preferred.

Example

There are numerous examples in current architectures where locator entities are used for keeping track of the location of mobile entities. In the Internet, a domain name system (DNS) [RFC1034] is employed to keep track of Internet hosts and the services on these hosts. In order to contact a server in the Internet, a calling entity needs to query the DNS first. The DNS name of the server serves as the unique identifier for the server entity. The response from a DNS for that query contains the location of that server, which can be used by the calling entity to contact it. The DNS is a distributed database where it takes a considerable period of time before location changes are disseminated through to all the cached copies that are maintained in the Internet. In fact, the DNS cannot be recommended for locating mobile entities (hosts or servers) that change location very frequently. For Mobile IP, in order to manage mobile nodes that can change location as frequently as once per second, a special architecture is employed that uses the Mobile IP protocol.

Background

Apart from the Internet, where the DNS service is used as a locator entity, other architectures have adopted similar schemes. The Common Object Request Broker Architecture (CORBA) [CORBAv2] employs a Naming Service with which server objects can register their location, in terms of an object reference. The location of a server object can be obtained by a client object, which can subsequently perform an operation on that server.

The Telecommunications Information Network Architecture (TINA) [TINA-SA5] [TINA-SAA5] specified a Naming Service that allows the computational objects of this architecture to register their interfaces. Other computational objects can use this service to obtain references to the interfaces of computational objects that they wish to call. The naming service of TINA was very similar to the CORBA naming service. In fact, the CORBA naming service has been adopted for most implementations of TINA to-date.

4.2.2 Mobile Entity Agent (MEA) Design Pattern

This design pattern tackles both problem areas that mobility introduces in the FoT: how mobile entities can be located, and how to cope with the unavailability of a mobile entity when it is changing location. Here follows the design pattern specification in the CORBA design pattern template [MM97]. A description of the CORBA design pattern template is given in Appendix A.

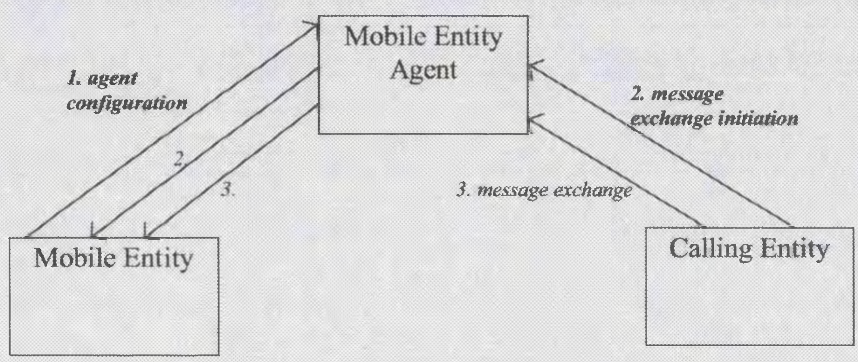
Most Applicable Scale: Enterprise

Solution Type: Technology

Solution Name: Mobile Entity Agent (MEA)

Intent: To outline how a mobile entity can be delegated for by a computational entity in a communications environment. This agent can act on behalf of the mobile entity while the latter is unavailable or unreachable.

Diagram



Primal Forces: Management of Functionality, Complexity and Change.

Applicability at This Scale

1. A mobile entity can be unavailable or unreachable while changing location.
2. A mobile entity delegates call handling to another mobile entity.
3. Only authorised entities can obtain, if desired, the location of a mobile entity.
4. A mobile entity can change location even during a message exchange with another entity.

Solution Summary

- For each mobile entity, provide a mobile entity agent, which delegates for the mobile entity. The presence of a mobile entity in a communications environment is manifested and delegated for by its agent. There is a single instance of a mobile entity agent active per mobile entity, at any given time.

- The mobile entity agent can handle calls from calling entities to the mobile entity.

- There are two steps for a calling entity to have a message exchange with a mobile entity. The first step is the call initiation, where the calling entity contacts the agent of the mobile entity asking to have a message exchange with the mobile entity. Depending on call handling decisions by the agent, the calling entity proceeds with the message exchange with the mobile entity via the agent.

Benefits

The agent is always available, even while the mobile entity is unavailable or unreachable.

The agent could also cache requests while the mobile entity that it delegates for is temporarily unavailable.

The mobile entity can enable call handling on its agent.

The mobile entity can enable access control on its agent.

The mobile entity has to update its location record with only one other entity, its agent.

The location of a mobile entity does not have to be known outside the agent.

Since the mobile entity agent acts as the only location of a mobile entity for all message exchanges, the calling entities do not need to specify the CEC that they belong too, when contacting the mobile entity (see section 4.1.3 on GID).

Consequences

There is processing overhead since there is an initiation phase before a message exchange takes place.

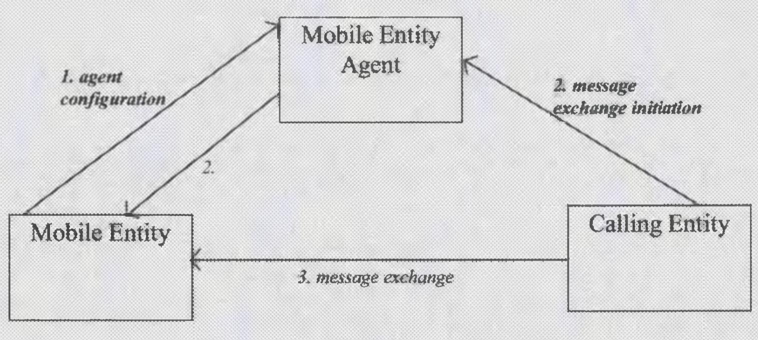
There is processing overhead since a message exchange has to go through the agent of a mobile entity.

Risk of triangular routing and increased network traffic (communications paths can be longer via the agent).

Since there is one agent per mobile entity, there is a need to locate the agent before the agent is contacted. This adds an extra stage before contacting a mobile entity.

If the Mobile Entity Agent of a Mobile Entity is unavailable, the Mobile Entity cannot be contacted, even if the Mobile Entity is available.

Variations to the Solution



One variation is shown in this diagram. In this case, the calling entity contacts the mobile entity agent first. Subject to call handling decisions, the agent returns the location of the mobile entity. Thereafter, the calling entity can proceed with the message exchange with the mobile entity.

If the mobile entity belongs to more than one Mobile Entity Class (MEC), the mobile entity agent will need the MEC identifier field of the Globally unique Identifier (GID) in order to determine its location. That is because a mobile entity can have different locations for different MEC and their corresponding Network Entity Classes (NECs).

This solution removes the benefit of keeping a mobile entity's location in the agent alone, but access control on who retrieves the entity's location can still apply. Also, this approach removes the consequences of triangular routing and processing overhead due to message exchanges directed via the agent. On the other hand, the location of the mobile entity may change after being retrieved by the calling entity. Also, this variation removes the benefit of contacting a mobile entity without having to specify the NEC that the calling entity belongs to (see section 4.1.3 on GID).

Rescaling to Other Levels

This pattern can apply at the global architecture level. On that level, the risk of triangular routing and locating the agent of a mobile entity become more important.

On the system or application architecture level, all the benefits of this design pattern may become less important due to the smaller scale.

Related Solutions

- The Agent Design Pattern, by T. J. Mowbray and R. C. Malveau. This design pattern simplifies client access to disparate information services and provides a uniform, consolidated access to disparate services.
- An expected OMG CORBAfacilities Design Pattern for agent technologies.
- The PROXY design pattern by E. Gamma et al [GHJV97], which provides a surrogate or placeholder for another object to control access to it.

- The ADAPTER design pattern by E. Gamma et al [GHJV97], which converts the interface of a class into another interface that clients expect.
- The DECORATOR design pattern by E. Gamma et al [GHJV97], which allows attaching additional responsibilities to an object dynamically.

Example

The concept of an agent that delegates for a mobile entity can be seen in the architecture that is suggested for the Mobile IP protocol in the Internet. Mobile IP aims to accommodate mobility for Internet hosts. The Internet hosts are given a unique IP address in the domain where they are initially registered; this is their home domain. A home agent can delegate for a number of mobile hosts. When a mobile host is at its home domain, it can be contacted in the normal way that Internet hosts are contacted. When the mobile host moves to a foreign domain, it has to register with a foreign agent first, and then it has to register its new location with its home agent. When a message is directed to the mobile host, the home agent will receive it on the mobile agent's behalf and it will forward it to the mobile host in the foreign domain by tunnelling.

N.B. the problem of triangular routing is one of the main drawbacks of the Mobile IP architecture. The location of the mobile entity is kept securely in the home agent. The mobile entity only has to contact the home agent to register its new location.

Background

Apart from the Internet, agents have been employed by the Telecommunications Information Network Architecture (TINA) [TINA-SA5]. TINA assumes that a number of service end-users can be attached to terminals in order to access telecommunications services. Both end-users and terminals are delegated for by user agents (UA) and terminal agents (TA) respectively. The UA keeps track of a person's location (terminal) and enforces call-handling rules on all calls directed to an end-user.

N.B. the term Intelligent Agent does not necessarily map to a mobile entity agent. Intelligent Agents can make intelligent decisions on behalf of the entity that they delegate for. A mobile entity agent contains the location of a mobile entity, it acts a mediator between callers and the mobile entity and it may make intelligent decisions (including access control and call handling) on behalf of it.

4.2.3 Combining the MEL and MEA design patterns

In the previous section, we presented the MEL and MEA design patterns, we discussed how they have been applied to current architectures and we investigated their pros and cons. We have observed that combining the two design patterns in a certain way can lift some of the cons of each individual pattern. This combination is presented in this section.

The approach we propose is shown in Figure 4-5 and features the following entities:

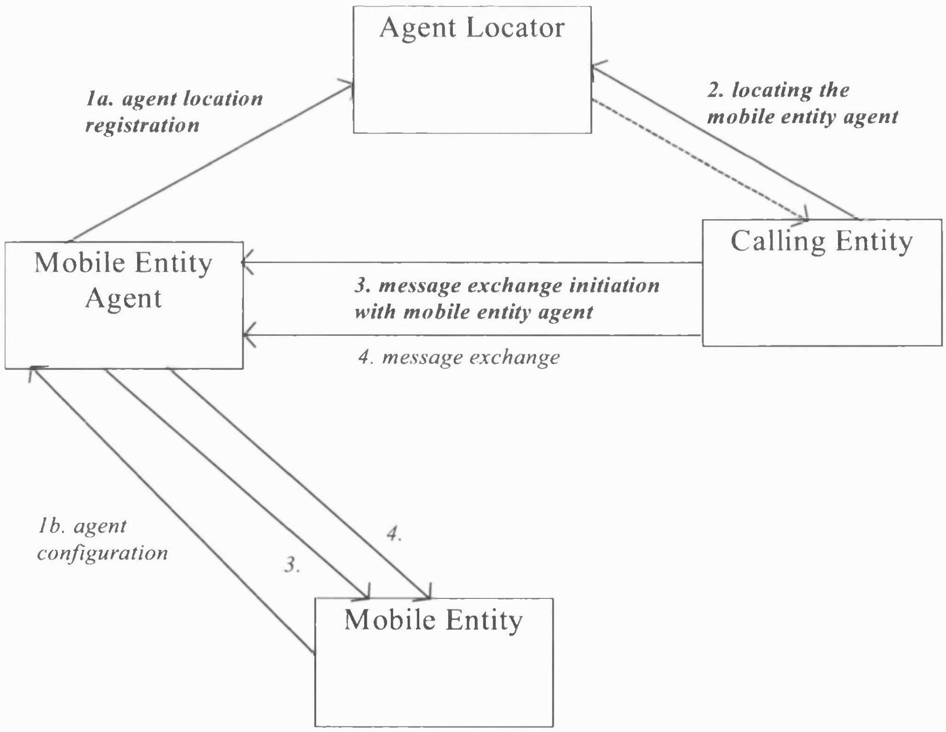


Figure 4-5: A recommended combination of the MEL and the MEA design patterns.

E1. *Calling entity*; is the network entity that calls a mobile entity.

E2. *Mobile entity*; is a mobile entity that can be contacted by other network entities.

E3. *Mobile entity agent*; is a network entity that delegates for the mobile entity, as the MEA design pattern defines. This agent has two important characteristics:

C1. It is either a fixed network entity or a mobile entity with a much lower average speed compared to the average speed of the mobile entity that it delegates for.

C2. It can be located via an Agent Locator entity as the MEL design pattern defines.

E4. *Agent locator*; is a network entity with a well known location to the calling entities, that keeps track of the location of the agents of mobile entities as the MEL design pattern defines.

In a scenario where a calling entity contacts a mobile entity, this arrangement of the two design patterns would work as follows:

S1. The mobile entity can configure its agent and register its location with its agent each time as in the MEA design pattern. Also, the agent registers its location (which is not expected to change often, if at all) with the agent locator as in the MEL design pattern.

S2. The calling entity contacts the agent locator and obtains the location of the mobile entity agent.

S3. The calling entity contacts the mobile entity agent and initiates a message exchange. Since the agent is expected to have a very low average and maximum speed, we expect that the location of the agent will not change during the time between steps (S2) and (S3).

S4. Subject to call handling decisions, the calling entity is allowed to proceed with the message exchange with the mobile entity.

We can see that certain consequences of the MEL design pattern are lifted in this way. First of all, since the locator is used to locate a low or zero speed agent, the retrieved location of the agent is not expected to become obsolete before the agent is contacted. Even if the agent locator is a distributed database, where it takes some time before all the cached copies of the location records are updated (like the DNS in the Internet), location changes of the agent are expected to be infrequent. That is because the agent is a low or zero speed entity. For the same reason, the impact of the agent being unavailable while on the move is minimised.

There is still the risk of triangular routing because message exchanges with the mobile entity go via the mobile entity agent. There are two alternatives to cure this problem. The first is to apply the alternative suggested in the MEA design pattern: the exchange is initiated via the mobile entity agent, and then, it is performed directly by the calling entity with the mobile entity. The other alternative is to move the mobile entity agent closer to the mobile entity in order to avoid triangular routing. For example, when the mobile entity changes domain completely (e.g. to a different country), its agent could move to the new domain with it. The agent does not have to move when the mobile entity changes location in the same domain.

From this analysis we see how certain drawbacks of the MEL and the MEA design patterns are cured in this combination of design patterns. However, it is not clear that this combination provides for better systems necessarily. This depends on the system that we apply these design patterns and on performance parameters.

This combination of design patterns has been applied in the European ACTS project Prospect for managing personal mobility and role mobility. In Prospect we used fixed mobile entity agents and the exchanges between the calling entities and the mobile entities were only initiated via the agents; these exchanges were subsequently performed directly with the mobile entities. The details are given in the following chapter.

5. Mapping the Framework of Terms to real environments

In chapter three we defined a Framework of Terms (FoT) for mobile systems. The FoT is intended to capture the common denominator of existing mobile systems. The deployment of the FoT was the first step of our approach as presented in the introduction. The second step was to investigate requirements for mobility on the FoT from two different ODP viewpoints: the information viewpoint and the computational viewpoint. We presented our results in chapter four: requirements regarding the necessity of unique identifiers for mobile entities, the scope of uniqueness of identifiers and a number of design patterns that can offer mobility support in the FoT.

In this chapter we present the third step and the fourth step of our approach. We map the FoT to real mobile systems and we relate the requirements of the FoT to the design of the real systems. This would allow us to investigate the generality of the FoT, its practical value and our claim that the FoT can assist to designing systems that avoid common problems of existing mobile system architectures as stated in our hypothesis.

In section 5.1 we present the design of a mobile system that was developed and demonstrated in the European ACTS project Prospect. We had the opportunity to design the system for *role mobility* support in Prospect. Through our involvement in the design, we had the opportunity to map the FoT to the Prospect environment and to make sure that the requirements of the FoT are respected in the role mobility system too. In section 5.2, we present how the role mobility system was implemented and demonstrated in the trial of the Prospect project.

In sections 5.3 and 5.4 we discuss how the FoT can map to the existing architectures of Mobile IP and GSM and we show where some of the FoT requirements are not respected. We discuss how these requirements could be avoided in the design of Mobile IP and GSM and how the problems that they currently have could be cured in this way.

5.1 A specification for Role Mobility in Prospect

The deregulation of the telecommunications market and the harnessing of new technologies have had an enormous impact on the telecommunications industry, leading to a dramatic increase in the number and type of services that telecommunications companies can offer. In this environment, mobility is becoming increasingly important. The success of mobile telephony indicates a demand from the telecommunications customers for increased flexibility in terms of service personalisation and mobility.

The Telecommunications Information Network Architecture Consortium (TINA-C) has been deploying a Service Architecture (SA) that defines how a range of diverse and possibly complex services can be provided and managed in a uniform way. The starting point for the TINA SA is a set of separations, which, among other aims, attempt to allow for mobility.

Much of the work done on mobility to-date is concerned with personal and terminal mobility. That is the case for TINA [TINA-overv, TINA-req, TINA-SA5, TINA-SAA5], ITU [F.850], ETSI [GSM03.02] and others. However, other kinds of mobility can be identified and tackled in a way similar to those for personal and terminal mobility. Role mobility is a problem that offered an interesting challenge to approach the concept of mobility from a new perspective.

5.1.1 The service environment in Prospect

The main purpose of Prospect was to demonstrate service management by means of a trial of a tele-education service (TES). This high level service provides courses to the students of a customer organisation who have been authorised to access it from the establishment of a contract between the TES provider and a TES customer. The TES is based on an integration of multi-media tele-services (MMTS) [LTMR97]: a multi-media conferencing service (MMC), a WebStore service allowing to store/retrieve information from a Web server, and a Hyper-Media service providing functionality above a Web server. The Prospect Management system is based on the one-stop-shopping paradigm, therefore the Customer needs to subscribe and interact only with the TES provider in order to receive the multi-media tele-services.

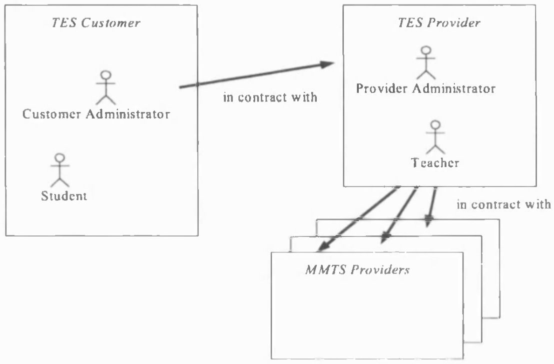


Figure 5-1: Prospect Enterprise Model.

The Enterprise Model of Prospect features a number of stakeholders with contractual responsibilities to each other (see Figure 5-1). The following actors and systems are employed to carry out the contractual responsibilities between stakeholders.

S1. The TES Provider system, which provides control and management for the TES.

S2. The TES Provider Administrator, who is responsible of the administration of the TES.

S3. The Teacher, who provides the content of the course and delivers the course.

S4. The TES Customer Administrator, who is responsible of the Customer domain management regarding the TES.

S5. Students, who attend the course.

S6. The MMTS Providers systems, which provide control and management of the MMTSs.

The service management functionality of the service providers' systems in Prospect includes subscription and accounting management. During the last phase of the Prospect trial the TES service management has been enhanced with a Role Management subsystem.

The Prospect components for service control and service management are based on the TINA Service Architecture [TINA-SA5, TINA-SAA5]. The implementation of the

Prospect environment makes use of the Orbix implementation of CORBA [CORBAv2] and the CORBA naming service [CORBAsvc].

5.1.2 Service roles and role mobility

In this section we define the concepts that relate to role mobility. We present our concept of a service role (or *role* for short), the location of a service role and how a service role can be modelled as a mobile entity. We also outline the areas of role management and role mobility.

5.1.2.1 Concept of a service role

Existing role theory [BT79] defines a role as a set of rights and duties. Lupu and Sloman [LS97, LMSY95] assume a paradigm where there are a number of positions in an organisation's hierarchy; the persons that occupy these positions are assigned to roles. These roles involve access rights to a set of target managed objects. With this scheme, Role-Based Access Control (RBAC) can be enabled to grand access for the persons that hold these roles to a set of managed objects.

On the other hand, Hamada points out some differences between roles in the traditional RBAC and roles in a telecommunications environment like TINA [Hamada98, Hamada97]. Hamada suggests that roles in TINA are service-based, rather than organisational. Also, he observes that roles in TINA are dynamically activated and bi-directional (there is a role-to-role association), while in RBAC they are static and uni-directional (they facilitate a role-to-object association). Apart from this, according to Hamada, roles are service-based and they are bound per TINA service session, while in RBAC roles are bound to a position within an organisation.

We consider our work complementary to the work by Hamada. We do not investigate the access rights of roles but the duties and the responsibilities of roles with regard to service provisioning. Also, we do not look at dynamically created roles, which Hamada does [Hamada97]. We assume that for every service a number of roles are created statically (e.g. "help-desk", "service administrator", etc.), and, when a user requests to contact a role, an instance of staff or user that will implement this role is determined dynamically. For this purpose, our role management component offers a brokerage service for contacting roles and it uses an activation model in order to accommodate as many instances of each role as required.

Therefore, we consider *service-roles* with the following characteristics:

C1. A service-role expresses *contractual responsibilities* between a service provider and a customer.

C2. A service-role involves a number of *access rights* and *duties* that derive from the contractual responsibilities between service provider and customer.

C3. A role can be taken on by a person, the *role-holder*. This person can either be staff of the provider organisation that have certain access rights to service resources[4] and duties to the customers, or persons from the customer organisation that have access rights to service resources and maybe duties to the provider organisation.

C4. We consider that a *role is assigned to a person (role-holder)*. This is an approach different from others [LS97, LMSY95], where persons are assigned to roles and inherit the access rights of a role. In our case, a role can also imply a set of duties and tasks. These tasks are implemented by role-holders. When a person (role-holder) takes on these tasks, we consider that this role is *assigned* to them. When a role is assigned to a role-holder, the role-holder, apart from the tasks, receives certain access rights too.

C5. Certain roles may require that more than one instance of a role is active at a given time. For example, a "help-desk" role of a service provider organisation may require that more than one role-holder is available at a given time to service for this role. Therefore, in order to control the instances of role in a service environment dynamically, *an activation model* is required.

C6. There can be *service independent roles*, like the "service administrator" role or the "help-desk" role. There can also be *service dependent* roles like the "doctor on call" role in a medical advice service, or the "tutor" role in a Tele-education service. Service independent roles can have well-defined and well-known rights and duties. If, for example, a service provider offers a "help-desk" role, the customers will know what to expect when contacting this role, without knowing the particular details of the service. For service dependent roles, the service provider must offer a description of their access rights and duties to the customers.

[4] Such resources may involve service control and service management.

We consider that every service provider needs a number of service roles. We believe that as the number and type of services on offer diversify, so the need for better and more effective service support and service provision grows. The concepts of service roles and role management can assist service providers to fulfil their obligations to their customers for service support and service provisioning more effectively. As such, we believe that a service provider must not only support person-to-person communication, but also role-to-person and role-to-role communication.

In this thesis we describe the specification of a loosely-roled system that was developed in the European ACTS project Prospect (contract number AC052). The term *loosely-roled system* is to relate this work to the concept of a *strongly-roled system* that is used by Hamada [Hamada97] to describe systems where only users that have been assigned to roles can participate in service sessions.

In Prospect, we focused on the duties that a role involves, rather than the access rights. We also assumed service roles that are implemented by the service provider; i.e. the role-holders are staff of the service provider. We provided a role management component, on the service provider's domain, that manages the life cycle of roles and their dynamic activation and we enhanced the service control part of Prospect, which is based on the TINA Service Architecture, to enable users to contact roles. At this stage, we did not provide for roles to contact users, or roles to contact roles although these cases are considered in the design of the component for role mobility support.

5.1.2.2 Role management and role mobility

A mobile entity is a well-known feature of most topics to do with mobility in telecommunications. It can change location without any disruption to the communication environment or to the service on offer. We consider the location of a mobile entity to be another entity via which the mobile entity can be reached. For example, personal mobility features a person that is able to change location. In this case, *the location of a person is a terminal* [F.850]. Via this terminal a person can be reached. Similarly, terminal mobility features a terminal that is able to change location. In this case, *the location of a terminal is a network access point* via which the terminal can be reached.

The concept of role mobility is not evident when we consider only the access rights of the roles and the role-based access control. However, when we consider the duties and the responsibilities of a role, and the requirement for dynamic activation of roles, the aspect of mobility is increasingly important. For example, let us consider the "help-desk" role for a multimedia conferencing service. This role, apart from access rights, has a number of responsibilities, like responding to users of the service who are having technical problems. The role of the service administrator may have to be available on a 24-hour basis in order to respond to such queries by the users. Let us assume that this role is assigned to one person (role-holder) at a time. Since the role will have to be available on a 24-hour basis, the person that the role is assigned to will have to change every few hours. We can consider that this role will have to "move" from one person (role-holder) to another during the course of time. Also, it may be the case that more than one role-holder will have to be available to service for this role at a given time. In this case, we can consider that more than one instance of a role can be active at a given time.

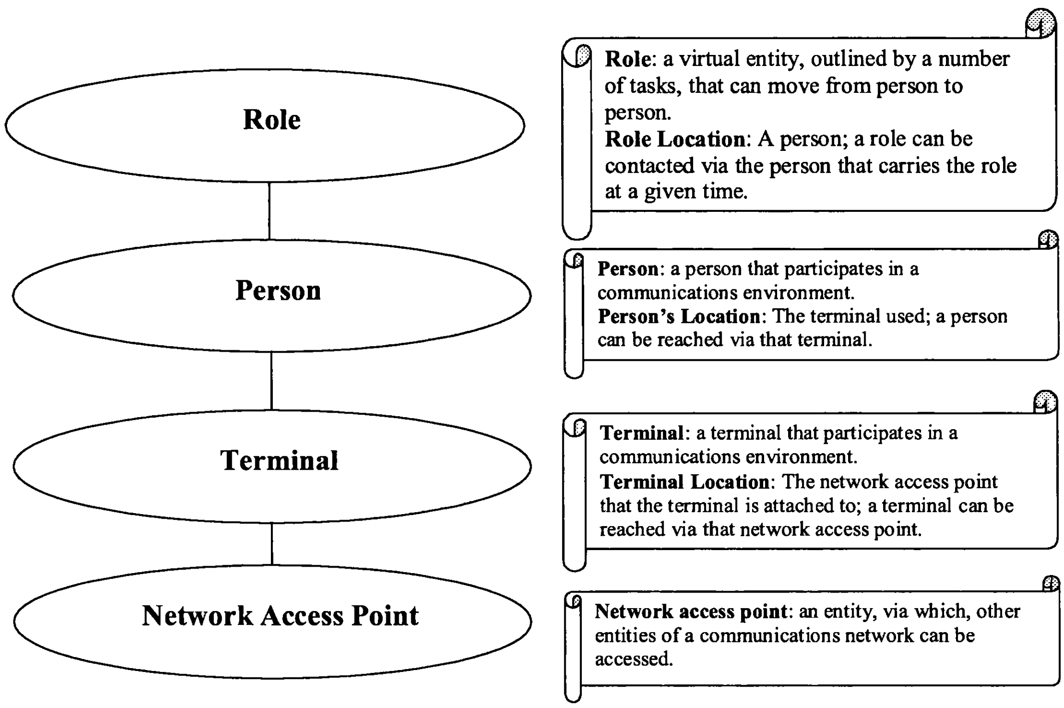


Figure 5-2: Example of network entities and their locations including roles.

From this viewpoint of role activation, we can perceive a role as a mobile entity that can be instantiated on one or more persons (role-holders). *The location of a role is the role-holder* (the person that takes on the role). During the course of time a role can move from one role-holder to another. In order to contact a role, a role-holder has to be determined first. In this way, we see that it is via a role-holder (the location of a role) that the tasks of a role are provided.). In Figure 5-2 we show an example different mobile entities such as roles, persons and terminals that a communications environment may feature. We also show how the location of these entities can be perceived.

The management of role activation, in this sense, can be considered a kind of mobility management. In the modelling process of the role management component in Prospect, we used the same techniques that were employed for managing personal or terminal mobility. For example, we used unique IDs to identify roles in the service provider's domain and we nominated an entity, a Role Agent to delegate for a role at all times, in a way similar to user agents or terminal agents.

The difference between role mobility and other kinds of mobility is that multiple instances of a role can exist at a time. A service role encapsulates a set of obligations and tasks, and it makes a virtual entity that can be attached to a number of role-holders. N.B. a service role is not a mobile agent, and role mobility does not imply agent mobility.

> *Treating a role as a mobile entity offers the advantage that we can reuse experience in the design of existing mobile systems that feature mobile entities such as persons or terminals.*

Also, it allows us to get an insight into the problem of mobility, since role mobility features a *virtual* mobile entity, unlike other kinds of mobility where the mobile entity is physical (i.e. a person or a terminal).

> *The role, as a "virtual" mobile entity, can be outlined by a set of tasks that it performs and it can be carried by persons, which act on its behalf. This "virtual" mobile entity can migrate from person to person.*

5.1.2.3 Role activation and role management

In this model we distinguish between different kinds of role-holders, depending on their activation status:

R1. A role can be assigned to a number of persons at the same time. A person who can fulfil a specific service role we call a *role-holder*.

R2. A role-holder that is able to fulfil the role when requested is known as an *active role-holder;* there may be many role-holders that service a role at any given time.

R3. A role-holder that is not ready to fulfil a role at a given time we call *inactive role-holder*.

R4. A role-holder that is in a service session with a user at a given time we call *role-holder in session*.

The role management system that we provided attempts to cope with the temporal aspects of role creation and role activation. For this purpose we have identified three areas of functionality in a role management system:

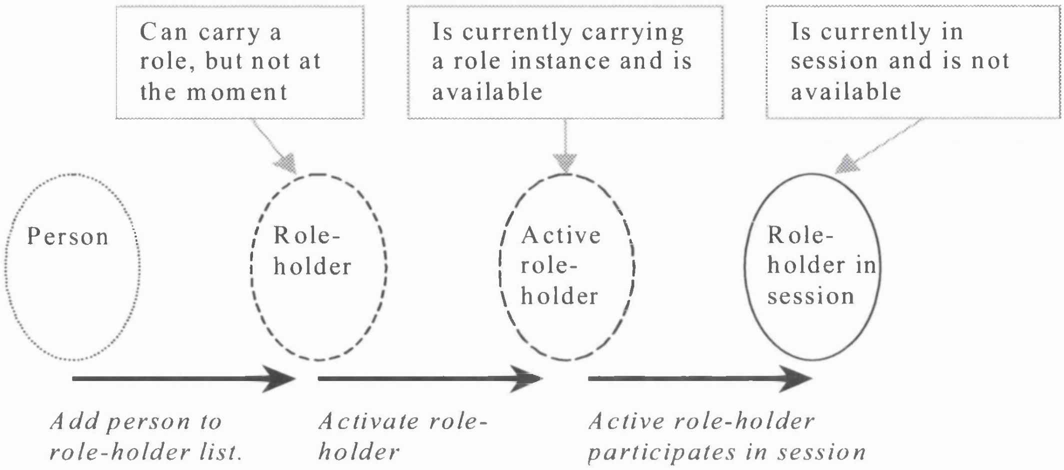


Figure 5-3: States of a role-holder.

A1. *Role life-cycle management*: is concerned with creating/deleting/updating service roles.

A2. *Role-holder management*: provides for adding/removing role-holders for a service role and for activating/deactivating role-holders Figure 5-3 shows the three states that a role-holder can be in. The activation/deactivation of role-holders can be policy based.

A3. *Call handling*: manages the forwarding of a call to a role. For example, when a call is addressed to a service role, the role management system decides which of the available active role-holders should service the call. Call handling can be policy based.

5.1.3 Mapping the FoT to the Prospect environment

The Prospect communications environment can be perceived at different levels from different viewpoints. According to the definitions of the FoT, each view of a communications environment defines a communications network (see section 3.1). In this section we chose one high-level perspective of the Prospect communications environment which features all the mobile entities involved. According to this view, we have the following entities in the communications network:

E1. *Terminals*; entities via which communications services can be accessed. Terminals are fixed entities in Prospect. Each of them has a unique identifier. The IP address of a terminal serves as its unique identifier. The scope of uniqueness of IP addresses is global in the Internet.

E2. *Users, or Persons*; they can be reached via the terminals that they are using. Persons are mobile entities; they can move from terminal to terminal.

E3. *Roles*; these are entities that can be reached via different users at different times. Roles are mobile entities: they can move from person to person. Examples of roles are a service administrator role and a private tutor role.

All the entities of this communications network, which appear in Figure 5-4, are connected via unicast of broadcast media between them. We can distinguish between the following Mobile Entity Classes (MEC):

MEC1. *Persons MEC*. Personal mobility is supported in Prospect. Persons can be contacted by other persons via the terminals to which they are attached. All the Persons share the same Absolute Location Space (ALS). The ALS of the Persons MEC is the set of terminals in the communications network. The Network Entity Class (NEC) that corresponds to the Persons MEC is again the set of persons of the communications network. The Complete Entity Class (CEC) that corresponds to the Persons MEC is all the Persons and Terminals of the communications network.

MEC2. *Roles MEC*. Role mobility is supported in Prospect. Roles can be contacted by persons via the persons that carry them (the role-holders). All the Roles share the same Absolute Location Space (ALS). The ALS of the Roles MEC is the set of persons in the communications network. The Network Entity Class (NEC) that corresponds to the Roles MEC is the Persons of the communications network. The Complete Entity Class (CEC) that corresponds to the Roles MEC is again the set of persons of the communications network. N.B. In Prospect we allowed only for persons to contact roles; we did not allow roles to contact other roles. For this reason, the Roles CEC in Figure 5-5 does not include the Roles.

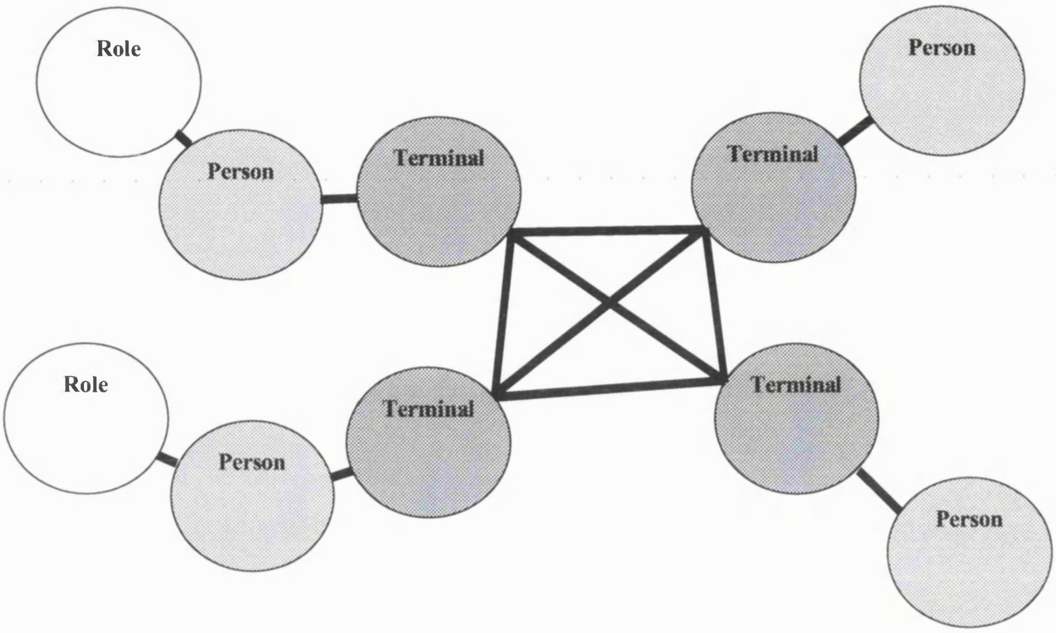


Figure 5-4: One communications network in Prospect.

Figure 5-5 illustrates the two MECs of the Prospect environment, together with their corresponding CECs. We also give examples of communication paths (CPs) than can be established to convey message exchanges between network entities of this communications network.

5.1.4 A specification of a role management component

In this section we give a high level description of a role management component. First we give the use cases that outline the functionality of the component and then we look at certain design considerations with respect to the information and computational viewpoints. The role management component, which also supports role mobility, was implemented and demonstrated successfully in the European ACTS project Prospect according to this specification. More details about the implementation of this component are given in section 5.2 of this thesis.

5.1.4.1 Use cases for the role management component

The functionality for the role management component can be described with a number of use cases [Tirop98]. The entities that interact in these use cases are:

E1. *The end-user*: who is the end-user of a service.

E2. *The role management system*: which is the computing facility that manages roles. The role management system resides on the service provider domain.

E3. *The role-holder*: who is a user who can carry a role.

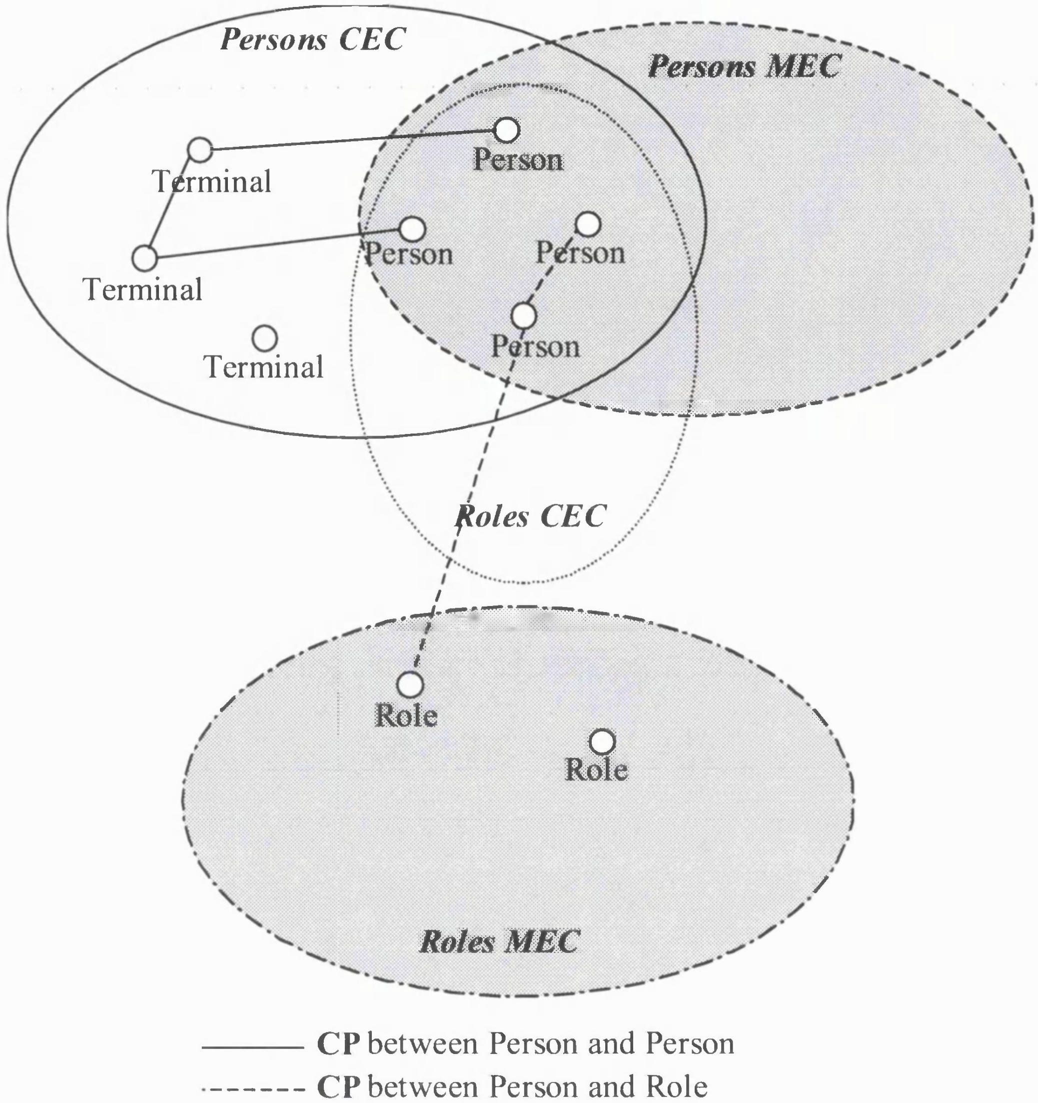


Figure 5-5: MECs and their corresponding CECs in a communications network of Prospect.

We also distinguish between *active role-holders* and *inactive role-holders*, as described in section 5.1.2.2 and as illustrated in Figure 5-3.

The use cases for role management and role-based exchanges are divided into four groups:

G1. *Role management use-cases that are initiated by the provider service administrator*: These provide for creating, updating or deleting roles. They allow the service administrator to add or remove role-holders for a particular role and to set-up policies for determining when a role-holder is activated (becomes available for calls), or deactivated (becomes unavailable for calls) or which of the active role-holders is to take a particular call to the role.

G2. *Role management use cases that are initiated by the role-holder*: These use cases allow a role-holder to become active (available for calls) or inactive (unavailable for calls), at will.

G3. *Role browsing*: Use cases that allow the service provider administrator, end-users or role-holders to browse through the roles of a service provider and get details about them.

G4. *Call use cases*: These allow an end-user to call an instance of a role. In other words, they allow an end-user to contact an instance of a service role carried by a role-holder. Also they allow for a role-holder to make a call to a person as the role; for example they allow a person (role-holder) who is carrying the service administrator role to make a call to an end-user as the service administrator.

In Figure 5-6 we illustrate the use cases that were considered in the design of the role management and role mobility system in Prospect in UML notation.

The use cases that relate to *role mobility management* are:

UC1. *Browse roles*: describes the exchange where a listing of roles and their attributes is returned for all the roles in a service provider domain.

UC2. *Set-up role*: describes the creation and initialisation of a role. An initial list of role-holders is set-up with the role.

UC3. *Update role*: describes the modification of the profile of a role. For example, modifications to policies for the activation of role-holders. There are two use cases that inherit from this one:

- **UC4.** *Add role-holder*: describes the exchange between the role-management system and a user who is to be registered as a role-holder for a role.
- **UC5.** *Remove role-holder*: describes the exchange between the role-management system and a role-holder that is to cease being a role-holder for a role.

UC6. *Delete role*: describes the deletion of a role from the role management system.

UC7. *Activate role-holder*: describes the exchange between the role management system and an inactive role-holder in order for the later to become active.

UC8. *Deactivate role-holder*: describes the exchange between the role management system and an active role-holder in order for the later to become inactive.

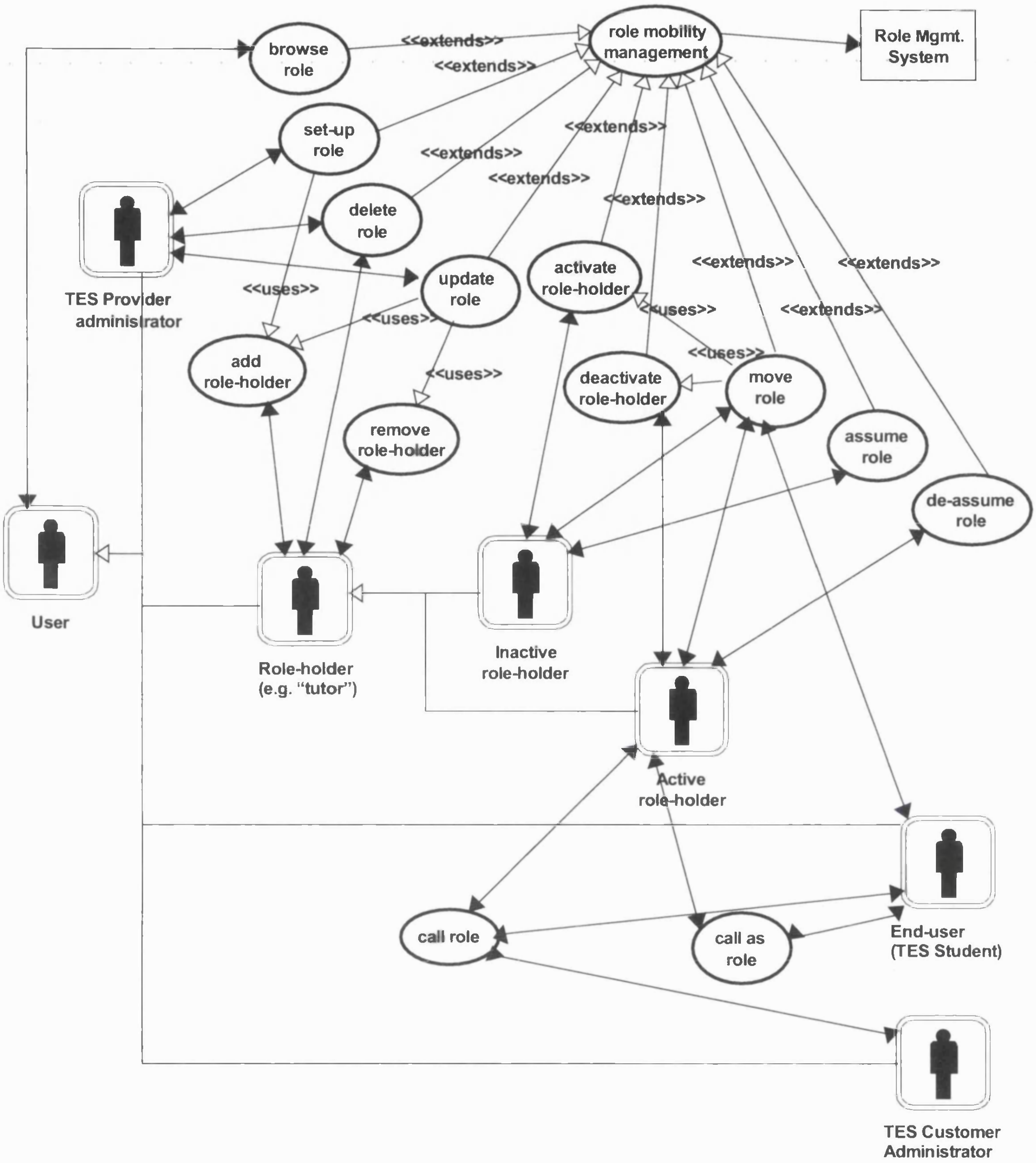


Figure 5-6: Role mobility and role management use cases.

UC9. *Move role*: describes the exchange between the role management system, an active role-holder and the users who are in a session with this role holder in order for the role to move to another role-holder.

UC10. *Assume role*: describes the exchange between the role management system and an inactive role-holder in order for the latter to become active.

UC11. *De-assume role*: describes the exchange between the role management system and an active role-holder in order for the later to become inactive.

There are two call related use cases;

UC12. The *call role* use case describes the exchange between an user/role-holder, a role-holder, the role management system, and other components of the call-model in order for an user/role-holder to contact a role.

UC13. The *call as role* use case describes the exchange between a role-holder, an user/role-holder, the role management system, and other components of the call-model in order for a role to contact an user/role-holder.

The above use cases set the requirements analysis for the design and deployment of the role management component. The information and computational aspects that derive from this analysis are presented in the following sections.

5.1.4.2 Information aspects

A role is identified by means of a role ID. Since the scope of a role is the service provider domain, the scope of the uniqueness of a role ID is this service provider domain. From the use cases of Figure 5-6 we can see that the following kinds of exchanges can take place in a communications environment with roles and end-users:

E1. A role (which is carried by a role-holder) contacts another role (on another role-holder).

E2. A role (which is carried by a role-holder) contacts an end-user.

E3. An end-user contacts a role (via the role-holder that is carrying that role).

E4. An end-user contacts another end-user.

The group of the different roles that are supported in a service provider's domain can be contacted by the end-users and the other roles of that domain. Similarly, the group of end-users of a service provider's domain can be contacted by the roles and the other end-users of that domain. This makes it clearer, that in order to address someone in this environment, we must be able to specify first whether it is a role or a user, and then to specify which exactly role or end-user is being addressed. The service control functionality supports this kind of addressing by means of a called party identifier that is composed of two fields:

<calledPartyIdentifier>=<calledPartyType><partyID>

The calledPartyType can either be *role* or *person* and the partyID identifies a person or a role *uniquely* in the set of persons or roles. It is the responsibility of the SP to make sure that a role ID is unique in the service provider domain. Table 5-1 shows these types of invitations in CORBA Interface Definition Language (IDL) structures [CORBAv2].

```
typedef short t_callPartyType;
const t_callPartyType t_user=0;
const t_callPartyType t_role=1;

union t_callParty
        switch (t_callPartyType) {
                case t_user:
                        t_userId user_id;
                case t_role:
                        t_roleId role_id;
};

struct t_invitation {
        t_callParty inviter,
        t_callParty invitee
};
```

Table 5-1: Information to support invitations and role-holder state management.

There is certain information that is associated with a role. This information includes the role ID, the list of the user IDs of the role-holders, the list of active role-holders and a number of policies regarding the activation/deactivation of role-holders and determining which of the active role-holders will take a call to the role. Therefore, a description of a role includes:

D1. The role ID.

D2. The list of the user IDs of the role-holders.

D3. The user IDs of the active role-holders.

D4. The policy for the activation/deactivation of a role-holder. For example, the time of day can determine when a role-holder becomes active.

D5. The policy for deciding which of the active role-holders will answer to a specific call. For example, the user ID of the caller can determine which of the active role-holders will take the call.

Table 5-2 gives the description of a role in CORBA IDL [CORBAv2] structures.

```
typedef string t_roleId;
typedef string t_userId;
typedef t_userId t_roleHolder
typedef sequence <t_roleHolder> t_roleHolderList;

typedef string t_criterionType;
typedef string t_criterionValue;

enum PolicyAction {
       activate
       deactivate
       call          // also, call_randomly, call_ordered, etc …
};

struct t_policyElement {
       t_criterionType type,          // e.g. time slot
       t_criterionValue value,        // e.g. <from-time>,<to-time>
       t_PolicyAction action,         // e.g. activate, call, etc …
       t_roleHolderList role_holders // e.g. roleholder A, roleholder C
};

typedef sequence <t_policyElement> t_policySpec;

struct role {
       t_roleId role_id,
       t_svcId service_id,            // e.g. the service the role is serving
       t_roleHolderList role_holder_list,
       t_roleHolderList active_role_holder_list,
       t_policySpec activation_policy,
       t_policySpec deactivation_policy,
       t_policySpec call_policy
};
```

Table 5-2: Role description.

5.1.4.3 Computational aspects

Figure 5-7 shows a computational model for a role management component, in UML notation. The rectangles denote computational objects; these map to UML categories. The circles denote interfaces.

According to the TINA USCM model (Universal Service Component Model) [TINA-SAA5] a computational object (CO) has four parts: usage, substance, management and core. The *usage* part supports the usage of a CO by other COs, the *substance* part allows the CO to use other COs, the *management* part allows for the management of the CO, and the *core* part implements the core functionality of the CO. The usage and the substance parts of the COs also appear in the diagram of Figure 5-7.

A role is a virtual mobile entity, which is outlined by a number of tasks. This means that the role is not a physical mobile entity like persons in personal mobility, terminals in terminal mobility or agents in agent mobility. Although it is not a physical entity, a role can still be delegated for by a *Role Agent* (RA) in a similar way like persons or terminals, which are delegated for by user agents or terminal agents respectively.

The RA has all the information regarding the role that it delegates for as shown in Table 5-2: role ID, list of role-holders, policy for activating/deactivating/calling role-holders. The RA contacts User Agents (UA) for the following use cases: add/remove role-holder (*i_uaRoleHolder*), activate/deactivate role-holder and move role (*i_uaRoleHolder*). A RA is contacted by UAs in the assume/de-assume role use cases (*i_raUsrMgmt*). There is only one RA per role in a service provider domain. The Role Agent can be contacted by the User Agent of the caller when a call is made to the role that the RA delegates for (*i_raContact*).

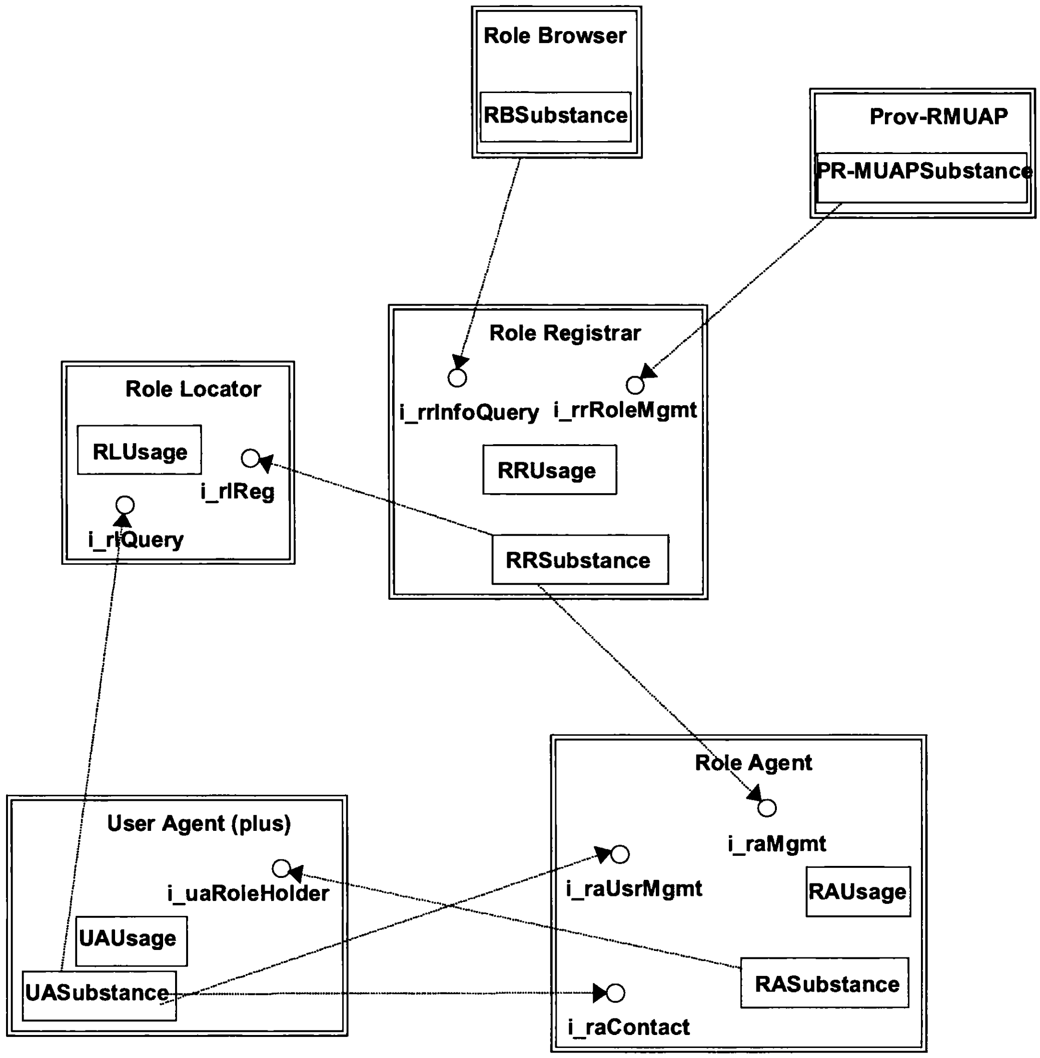


Figure 5-7: Computational view of the role management system.

The *Role Registrar* (RR) keeps track of all the RAs in a service provider domain. It manages the life cycle of a RA and it can give a list of all the roles provided by a service provider (*i_rrInfoQuery*). The role management user application of the provider administrator (Prov-RMUAP) and the role browser application interact with the RR in the following use cases: set-up/update/delete role (*i_rrRoleMgmt*), browse roles (*i_rrInfoQuery*). The Role Registrar manages a particular role via the RA that delegates for this role (*i_raMgmt*). There is one RR per service provider domain.

The *Role Locator* (RL) is able to provide the location of the RA for a given role ID. It interacts with the UA in the call role use case and the role call use (*i_rlQuery*). The RR also uses the RL in order to register a new RA in the set-up role use case (*i_rlReg*). There is one RL per service provider domain.

The use of a Role Agent to delegate for a role has certain benefits. The RA of a role can be always available even if there is no instance of a role carried by a role-holder at some moment. Also, The RA can handle calls to a role by means of a role profile and it can also enforce access control to accessing the role. The RA can initiate more instances of a role on demand, if there are many callers to the particular role. Finally, the RA keeps track of where the different instances of a role are available (the role-holders who are currently carrying the role).

Of course there are some drawbacks. Firstly, a caller has to contact the RA before they can get through to the role-holder and the role. Second, if the RA is not available, the role cannot be contacted even if it is instantiated on a number of role-holders.

Assuming that the users are delegated for by user agents and since the role-holders are users who are delegated for by user agents too, we make sure that the user agents of the role-holders support the necessary exchanges with RAs (for example for activating a role-holder).

Although a role is a mobile entity, the RA does not need to be mobile. In order to get in contact with the RA of a role we need to use another entity the Role Agent Locator, or *Role Locator (RL)* which can return the location of a role agent given a role ID. We note that the RL locates RAs and not roles.

A service provider can offer a number of roles. In order to keep track of all the roles in a service provider domain we employ a Role Registrar (RR). This entity manages the lifecycle of roles and RAs, i.e. it can create/update/delete RAs and it can return information about all the roles of a service provider domain.

5.1.5 Mapping the FoT requirements to the Prospect environment

In chapter four we identified information and computational level requirements on the Framework of Terms (FoT). We are now going to verify that the requirements that we identified in chapter four are respected in the design of the Prospect model.

5.1.5.1 Information viewpoint requirements

In Table 4-1 we summarise the information viewpoint requirements for mobile entities in the FoT. Here, we examine each requirement separately to see if they are satisfied in the design of the Prospect system for role mobility and personal mobility.

R1. Identifiers are used for both persons and roles in Prospect. Therefore, all the mobile entities in the Prospect environment are identifiable.

R2. The identifiers that are used for the roles are unique in the Roles MEC. Also, the identifiers used for persons are unique in the Persons MEC. Therefore, this requirement is satisfied too.

R3. The members of the Roles CEC (which is all the persons in the Prospect environment) use the same identifier for each role of the corresponding Roles MEC. Also, the members of the Persons CEC (which is all the persons and terminals in the Prospect environment) use the same identifier for each person of the corresponding Persons MEC. Therefore, this requirement is satisfied.

R4. In the Prospect environment, the entities of the Roles CEC are also members of the Persons CEC. Also, the members of the Persons CEC are also members of the Roles CEC. Therefore, this requirement does not apply for Prospect.

R5. Since entities from both the Roles CEC and the Persons CEC belong to more than one CEC, a global identifier is employed in Prospect. This global identifier is called "*calledPartyIdentifier*" and it consists of two fields: an MEC identifier and an entity identifier. The MEC identifier is described by the "*callPartyType*" field, which can take two values: *role* (for addressing members of the Roles MEC) or *person* (for addressing members of the Persons MEC). The entity identifier is described by the "*partyID*" field, which for persons is unique in the Persons MEC and for roles is unique in the Roles MEC. Therefore, this requirement is satisfied in Prospect.

5.1.5.2 Computational viewpoint requirements

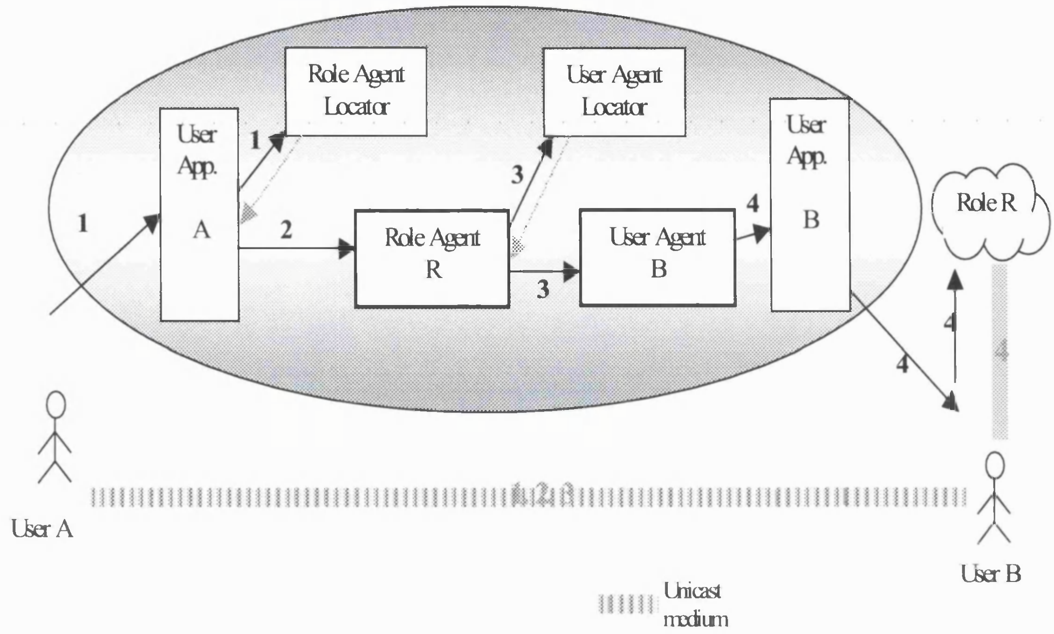


Figure 5-8: Exchange initiation between user and role in Prospect.

During the design of the TINA-like management system in Prospect we applied the combination of the MEL and MEA design patterns. Both mobile entities (persons and roles) were delegated for by a fixed user agent (UA) and a role agent (RA) respectively. The UA holds the location of a person (user), which is a terminal. The RA holds the location of a role, which is a person. We employ a user agent locator or user locator to locate the UAs of individual users. Similarly, we employ role agent locators or role locators to locate the RAs of individual roles [TMK99]. The diagram of Figure 5-8 illustrates an example of a scenario, where *User A* initiates a message exchange with (or makes a call to) *Role R*. This role is located at *User B* at that moment, therefore, the call is routed through *User B*.

From the users/roles point of view, the following steps take place when *User A* initiates an exchange with *Role R*:

S1, S2, S3: *User A* contacts *User B*, where the *Role R* is currently attached.
S4: The call is forwarded to *Role R*, which is served by *User B*.

From the management architecture's point of view, where the MEL and MEA combination was applied, the following steps take place:

MS1. When *User A* initiates the call, the user application of *User A* uses the role agent locator to obtain the location of the role agent of *Role R*.

MS2. The user application of *User A* contacts the role agent of *Role R*.

MS3. The role agent of *Role R* knows which person (user) is currently holding *Role R*, i.e. on which user the role R is currently located. Therefore, it contacts the user agent locator to find the location of the user agent of *User B* who is currently holding this role.

MS4. The role agent of *Role R* forwards the message exchange initiation to the user agent of *User B*. From this point, the call is forwarded to the user application of *User B*, then to *User B* and then to the role that user B implements, *Role R*.

In this scenario, we discuss only the message initiation phase (the call) and not the exchange itself. Only the initiation of exchanges takes place via the user agents and role agents. The user/role agents return references to the caller, which can be used by the caller to have an exchange directly with the entity called. This is the alternative that was suggested in the MEA design pattern in section 4.2.2.

Since the agents were fixed entities we had all the benefits and consequences of that. If the agent of an entity was unavailable, the entity could not have been contacted. We also had some processing overhead, since all calls to an entity had to go through the agent first. On the other hand, since only the message exchange initiation was performed via the agents, there was no risk of triangular routing.

An important assumption that we made in Prospect was that once a message exchange initiation takes place successfully, the called entity would not change location before the message exchange completes. This assumption lifted the consequences of locations becoming obsolete just after being retrieved and before the actual exchange takes place.

Since the agents are fixed entities, the only consequences of the MEL design pattern that still applied were the processing overhead of contacting the locators to retrieve the location of the agents and that if the locators were unavailable, no agents could be contacted.

5.2 An implementation of Role Mobility in Prospect

In this section we present the implementation of the role management and role mobility subsystem in the Prospect environment. This subsystem was demonstrated during the last trial of the European ACTS project Prospect (contract number AC052) in the premises of Tele-Denmark, in August 1998. Most of the text and the figures of this section is taken from a paper on "A generic component for managing service roles", by Thanassis Tiropanis, Chris Malbon and Hervé Karp [TMK99], which presents the role management component in Prospect.

5.2.1 Mobility modelling in Prospect

Prospect provided for personal mobility and for role mobility. Mainly, TINA mechanisms for personal mobility support were employed. User Agents (UA) were used to delegate for each user. Calls to a particular user were handled by the user's UA. The UA would register itself with the User Agent Locator, which in Prospect was the CORBA Naming Service. The user would register its location (the terminal user) with the UA. In this way, the UA maintained a binding between the mobile entity (the user) and its location (the terminal used). When a caller invited a user to a session, the CORBA naming service was used to locate the UA of the user. The UA would then be able to pass the invitation to the user, knowing the terminal at which the user was located.

Role mobility was modelled along the same lines with personal mobility. Role Agents (RA) were used to delegate for service roles. For every role, one instance of a RA was created. Calls to a particular role were handled by the role's RA. The RA would register itself with the Role Agent Locator, which in Prospect was the CORBA Naming Service. The RA maintained a binding between the mobile entity (the role) and its location (the role-holder that would service for the role)[5]. When a caller invited a role to a session, the CORBA naming service was used to locate the RA of the role. The RA would then be able to determine the location of a role (the role-holder to take the call) and it would pass the invitation to the User Agent of the appropriate role-holder.

---

[5] If required, the RA could create more instances of a role, i.e. activate more role-holders to take on the role.

5.2.2 Implementation use cases

A subset of the use cases of 5.1.4.1 was implemented and demonstrated in Prospect. The selection of the use cases to be implemented in the project was based on certain assumptions. A first assumption was that in most cases it is the end-user of a service that will have to contact a role. Therefore, the “call as role” use case was not given first priority for the implementation. Another assumption was that there would be no need to move a role from one active role-holder to another during a session between an end-user and the role. Therefore, the “move role” use case was not implemented in the Prospect environment.

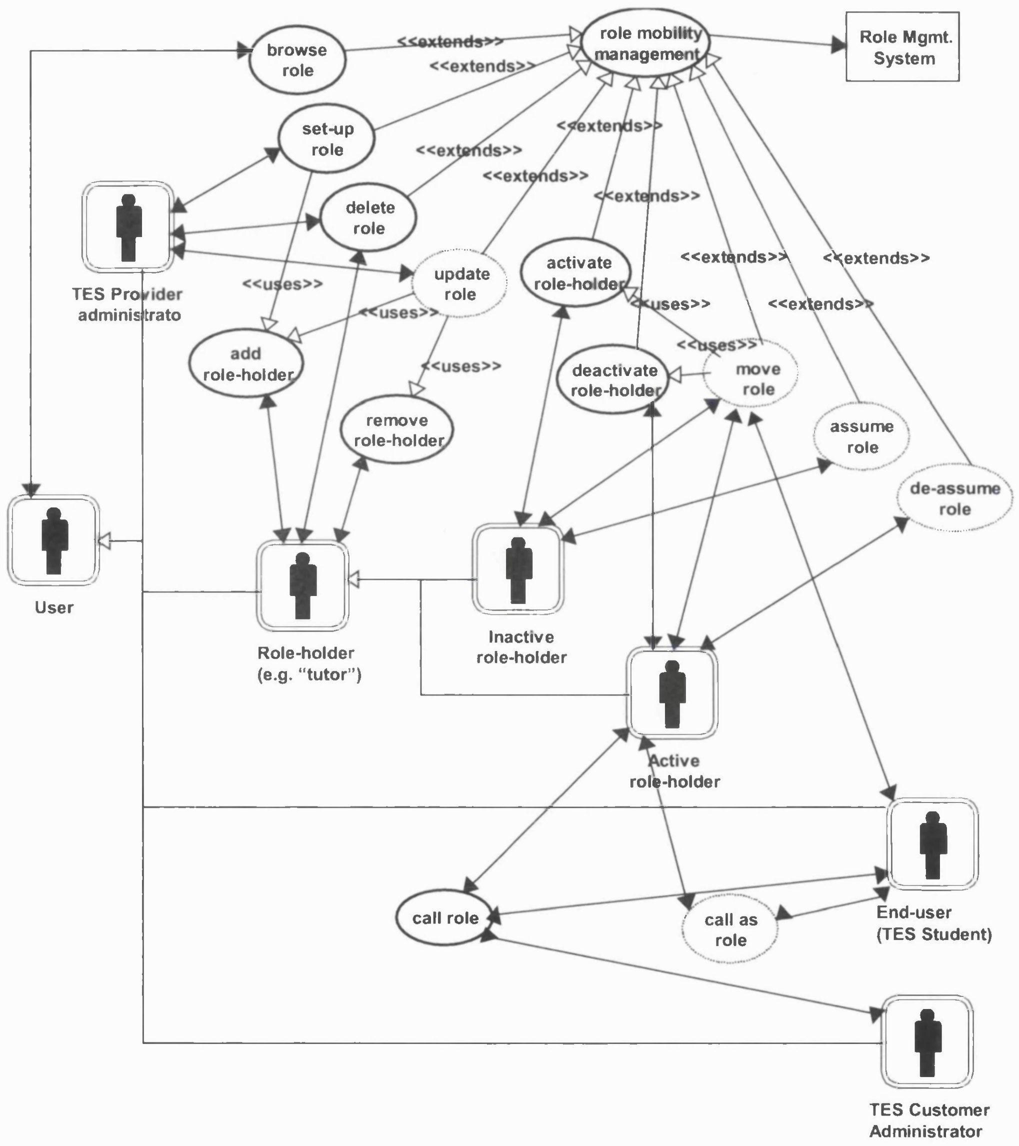


Figure 5-9: Role mobility and role management implementation use cases.

From the role management use cases that were initially considered, the ones that were eventually implemented and demonstrated are highlighted in Figure 5-9. These are:

IUC1. *Browse role*: the end-user receives a list of roles and can ask for details on a specific one.

IUC2. *Set-up role*: the service provider administrator creates a new role, including all role-holders and policies.

IUC3. *Delete role*: a role is deleted from the role management system.

IUC4. *Add role-holder*: a new role-holder is registered for a role.

IUC5. *Remove role-holder*: an existing role-holder is removed from a role.

IUC6. *Activate role-holder*: the service provider/administrator instructs an existing role-holder to become active (or the system activates a role-holder based on predefined policies).

IUC7. *Deactivate role-holder*: the administrator instructs an activated role-holder to stand down (or the system deactivates a role-holder based on pre-defined policies).

IUC8. *Call role:* a user calls a role (in our case, a student contacts the "tutor" role) and gets through to one of the role-holders who are carrying this role at that time.

5.2.3 Role Management sub-system

This section describes the implemented Role Mobility sub-system and includes descriptions of the following components: the Role Registrar (RR), the Role Agent (RA), the Role Locator (RL) and the User Locator (UL). It includes additions to the User Agent (UA), and a brief description of the Provider Management User Application (PMUAP) and the User Application (UAP). The components of the role management system and their interfaces were implemented in Java with CORBA support [CORBAv2]. A view of the components and their relationships is given in the diagram of Figure 5-10.

Role Registrar

The RR manages the life cycle of the roles of the service provider. RAs are manipulated through the RR's *i_rrQuery interface,* allowing roles to be browsed and queried, and *i_rrMgmt* interface. The *i_rrMgmt* interface is used by the PMUAP to create, modify and delete roles, to alter role-holder lists, to change role and role-holder policies, and to activate and deactivate role-holders.

When a role is created the management system instructs the RR to create a new RA. The RR assigns to the role a unique identification moniker, a role ID, which is used for unique identification. RAs are stored as a collection of objects. In the original design storage was managed by a Role Locator (RL), however, for expediency, we used our chosen vendor's implementation of the CORBA Naming Service to publish a reference to each RA object. Role Agent objects were stored persistently using Java's Object Serialisation interface.

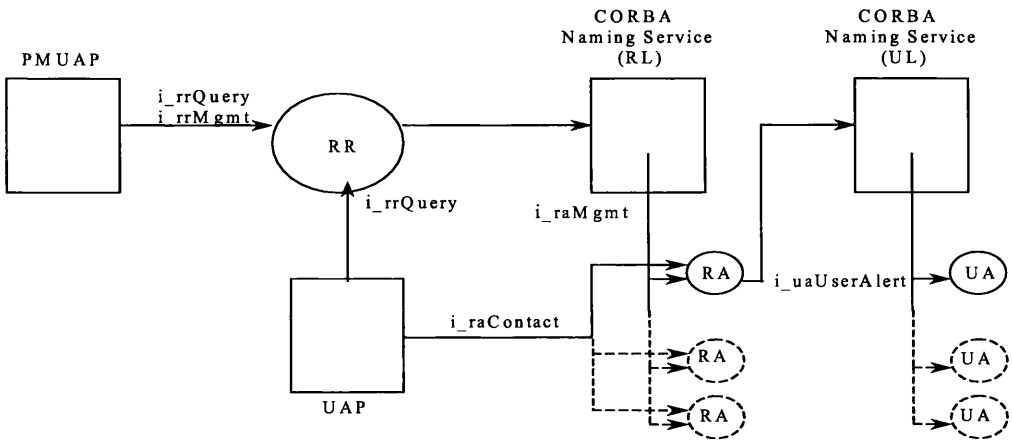


Figure 5-10: Component view of the role management system.

Role Locator

The RL is responsible for creating and publishing RA IOR's. Obviously there are security issues involved as it is unwise to make sensitive object information available to a wider community, however, in our framework the Naming Service ran locally within the provider's domain and was therefore as secure as any other information within the provider's domain.

Role Agent

The role agent contains a collection of role-holders that fulfil the requirements of that role. There may be one or more role-holders assigned to the role but only a subset of these will be available at any given time, these are known as *active* Role-Holders. The role-holders' activation status is maintained by the RA and is based on some policy. A policy is a rule that determines when a role-holder becomes active or not, and how an active role-holder is chosen when a request from a user is received.

Policies are set up through the PMUAP and the *i_raMgmt* interface, and can either be applied at the RA level, in which case they apply to all role-holders, or at the individual role-holder level when they apply only to that role-holder. It is the responsibility of the management system to ensure that, in the case of policy conflicts, one policy is preferred over another.

When a user wishes to contact a role, they do so, by calling the *i_raContact* interface of the required RA directly from the UAP. The sequence of events is as follows: the user asks the RR, by way of the *i_rrQuery* interface, for a list of supported roles, which the RR obtains from the Naming Service; this list contains the role-id, a role description and an IOR for the *i_raContact* interface for the RA of each role; once the user has selected the desired role, the IOR is narrowed and an invitation can be made to a role-holder.

To optimise the search, active role-holders are posted onto an Active Role-Holder List. Role-holders can notify the list when they become available. Proactive notification saves time searching for active role-holders on each request. Once an active role-holder is identified the RA locates the role-holder's *i_uaUserAlert* interface, an extension to the interfaces offered by the role-holder's UA (see below). The role-holder is informed of the request and invited to respond. If accepted, the call can commence. If rejected, another role-holder is selected until all active role-holders have been exhausted.

Role-Holder

Each role-holder maintains a policy list onto which new policies are posted and ordered. Policies determine when and how a role-holder becomes active.

Our implementation upheld three role-holder selection policies: selection based on priority, whereby a role-holder held some rank and was selected based on that rank, the higher the rank the more likely the role-holder was to being selected; selection based on preference, whereby a user could stipulate a preferred role-holder, if available this role-holder was requested above all others; and selection based on equality, whereby all active role-holders were placed in a queue, once a role-holder had finished servicing a call they were placed to the back of this queue.

Policy Element

A policy is an entity that operates at either the Role Agent level over all role-holders, or at the role-holder level.

Policy elements are created and/or modified at the PMUAP level before being assigned to a role-holder/role, and determine how and when role-holders active. Although policy was pre-determined, the use of a policy element allowed us the flexibility to create new policies.

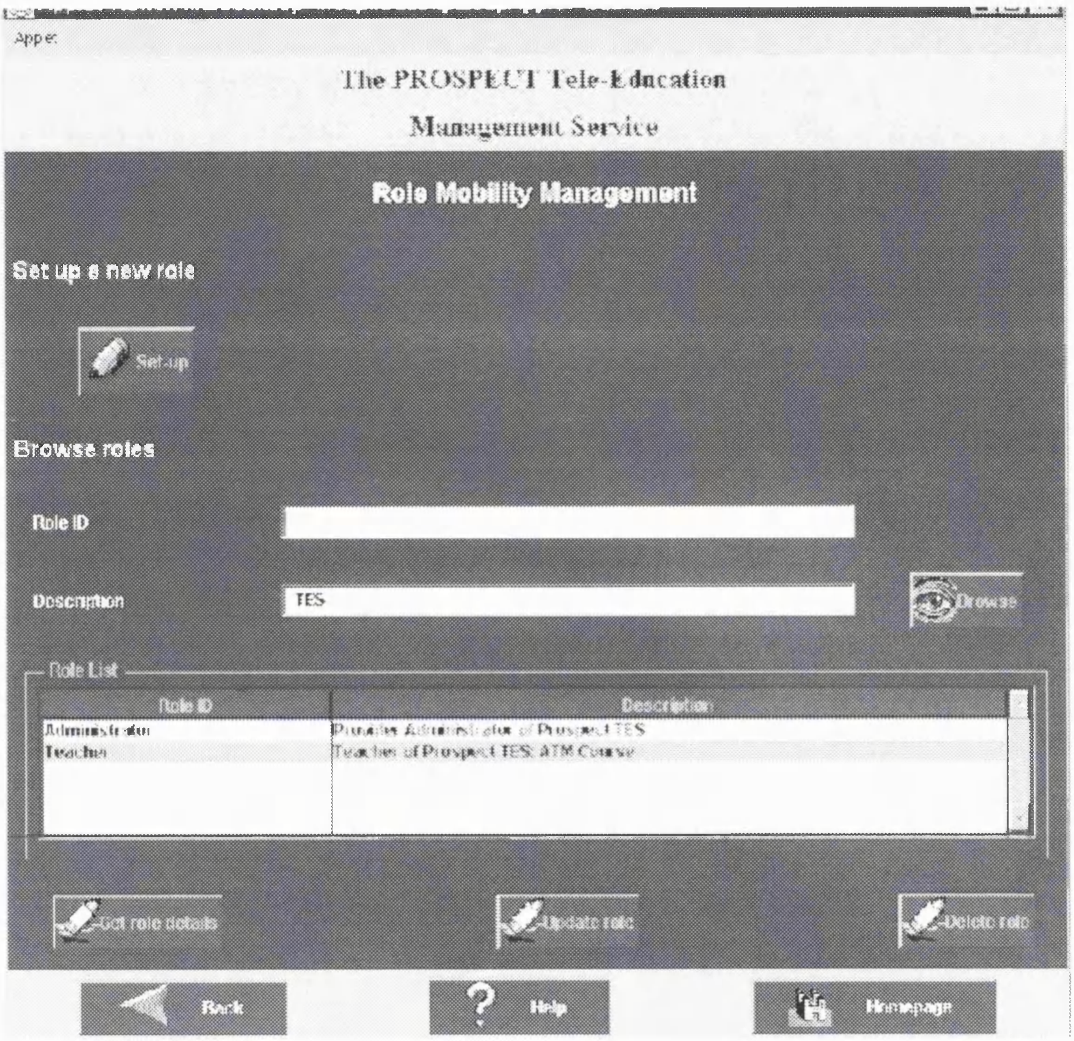


Figure 5-11: Main panel of the Provider Management UAP for Role Management.

User Agent Extensions

In order for a role-holder to physically be contacted by the RA or end-users the role-holder's UA required extending so that they could be notified of incoming requests for activation/deactivation/calls. The *i_uaUserAlert* interface enabled role-holders to be contacted in two ways: either by notification, requiring no role-holder interaction; or by invitation, requiring a response.

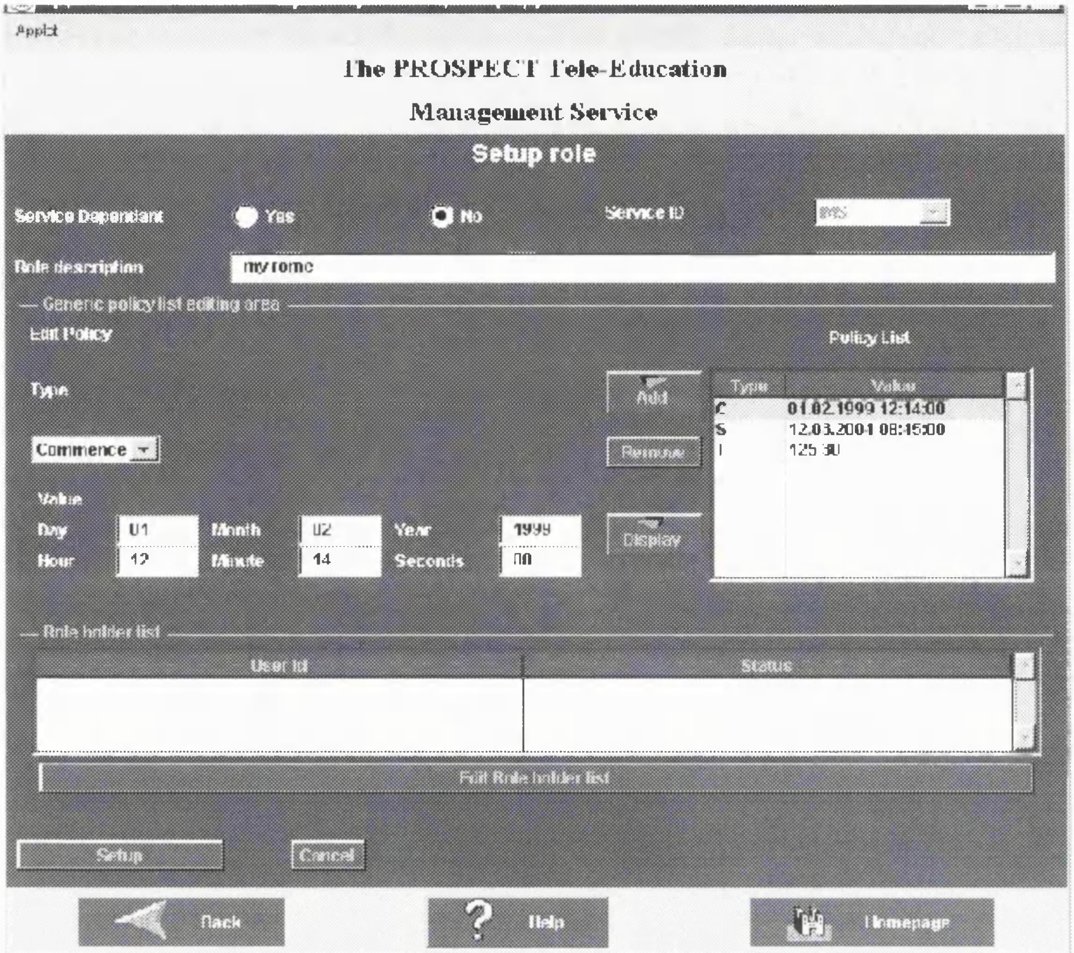


Figure 5-12: Panel of the Provider Management UAP for setting-up roles.

The design stipulates the use of a User Locator (UL) computational object to bring in contact a RA to the UA of a role-holder. A RA queries the UL using a known user ID which would return the appropriate interface. Again, for expediency, we chose to implement the UL using the Naming Service, publishing the *i_uaUserAlert* IOR using a defined naming convention.

5.2.4 The Provider administrator Management User Application (PMUAP)

The Provider Management UAP allows the Provider Administrator, in charge of the role management, to manage the information relative to the roles. The main functions provided are:

F1. *Browse role*, allowing to get a list of roles from user defined criteria (see Figure 5-11).

F2. *Set-up a new role*, allowing to create a role including a generic policy list and a role-holder list (see Figure 5-12).

F3. *Get role details*, allowing to get all the information relative to a role and details about each role-holder.

F4. *Update some role information*, allowing managing the role-holder list and activating or deactivating role-holders.

5.3 Mobile IP

In this section, we examine how the FoT can map to the Mobile IP architecture. In order to do that, we start from the mobile entities that the Mobile IP architecture defines and their location. We then determine the Mobile Entity Classes (MEC) that appear and their corresponding Complete Entity Classes (CEC). We are then ready to check if the requirements on the information and computational level are satisfied in the Mobile IP architecture. In this survey we examine the case of Mobile IPv4. However, we briefly mention how Mobile IPv6 [RFC1883] and Mobile IPv4 with route optimisation [PJ00] differentiate from plain Mobile IPv4 with respect to the information and computational level requirements.

5.3.1 Mapping the FoT to the Mobile IP architecture

The mobile entities that the Mobile IP architecture supports are IP nodes [RFC2002]. These nodes are located on IP networks or subnetworks and they gain connectivity via the IP routers of these networks.

When an ordinary IP node is contacted, the IP datagrams to the IP node are routed via the IP router of the network that the IP node is attached to, at a given time.

When a Mobile IP node is at the home network, IP datagrams are routed to it via the IP router of the home network. When a Mobile IP node is away from its home network, IP datagrams are routed to it via its Home Agent at the home network. The Home Agent (HA) is a special IP router that can forward IP datagrams to a given remote location that the Mobile IP node is at a given time. The remote location can be either a network interface that the Mobile IP node can attach to at the foreign network, or a special Foreign Agent (FA) at the remote network. In this survey we assume that a Mobile IP node is always attached to a FA when it is away from its home network. An overview of the operation of Mobile IP and the main elements of the Mobile IP architecture is given in section 2.1.

We select a view of the Mobile IP architecture that comprises the mobile entities (Mobile IP nodes) and their locations. This is our communications network according to the FoT. In this communications network, we have the following entities:

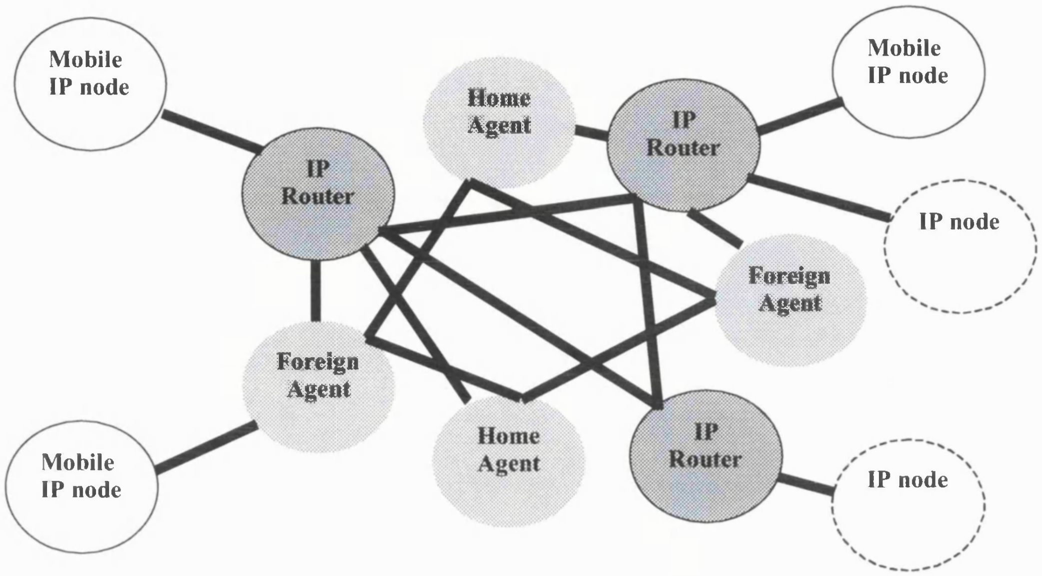


Figure 5-13: Example of a Mobile IP communications network.

E1. *Mobile IP node*. This is a mobile network entity. Note that a mobile IP node can also have router capabilities [RFC2002]. This case is not considered in this analysis.

E2. *IP router*. We consider the standard IP router a fixed network entity.

E3. *Home Agent (HA)*. We consider a HA a fixed network entity.

E4. *Foreign Agents (FA)*. We consider a FA a fixed network entity.

E5. *IP node*. This in our example is a fixed IP node, i.e. an IP node without Mobile IP support. N.B. in the Internet, an IP router is also an IP node. We assume that an "IP node" does not have routing capabilities in this analysis.

An example of a Mobile IP communications network that features these entities is illustrated in Figure 5-13. In this example all the entities are interconnected via unicast media.

The Absolute Location Space (ALS) of a Mobile IP node is a set that comprises: the IP router of the home network of the Mobile IP node and all the FAs to which the Mobile IP node can attach itself. Consequently, the Mobile IP nodes that are registered on the same home IP network and are served by the same IP router, share the same ALS. Therefore, *the Mobile IP nodes from the same home network/IP router make a separate MEC.*

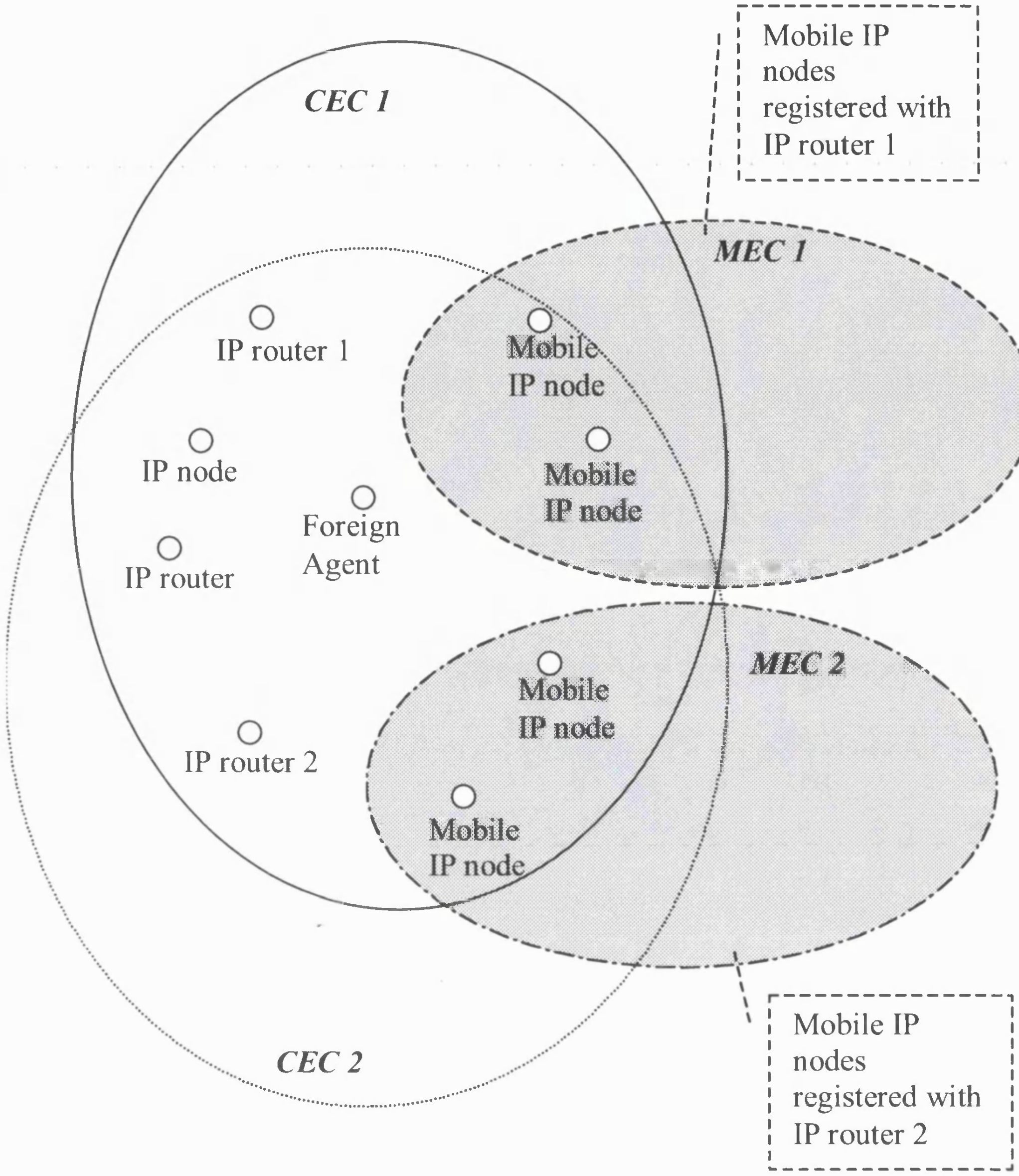


Figure 5-14: Example of MECs and corresponding CECs in a Mobile IP communications network.

The corresponding NEC of each MEC is the complete set of IP nodes, Mobile IP nodes, IP routers and Foreign Agents of the Internet. The corresponding CEC of each MEC is the NEC plus the ALS of each Mobile IP node, by definition. Therefore, the CEC comprises all the IP nodes, Mobile IP nodes, IP routers and Foreign Agents of the network; this makes the CEC identical to the NEC. We consider that every Mobile IP node can attach itself to any FA, and that there are not firewalls in our environment. Figure 5-14 illustrates two different MECs and CECs that we can have in a Mobile IP communications network. Note that all the MECs share an identical CEC in the Mobile IP communications environment.

Having mapped the FoT to the architecture of Mobile IP, we will now examine if the FoT requirements at the information and computational levels are satisfied in the Mobile IP architecture.

5.3.2 Information viewpoint requirements

In Table 4-1 we summarise the information viewpoint requirements for mobile entities in the FoT. Here, we examine each requirement separately to see if they are satisfied in the design of the Mobile IP architecture.

R1. Identifiers are used for Mobile IP nodes. The IP address of a Mobile IP node is used as an identifier. Therefore, this requirement is satisfied in the Mobile IP architecture.

R2. The IP addresses that identify Mobile IP nodes are unique. Therefore, this requirement is satisfied too.

R3. All the members of any CEC use the same identifier to identify the Mobile IP nodes of the corresponding MEC. Even the Home Agent of a Mobile IP node uses the IP address to identify it and to make an association between the Mobile IP node and the care-of-address that the node attached to. Therefore, this requirement is satisfied.

R4. As we pointed out in the previous paragraphs, all the MECs of a Mobile IP communications network share the same CEC. Since there are more than one home network in a Mobile IP communications network, there are more than one CECs and all of them have common entities. Therefore, this requirement does not apply for the Mobile IP architecture.

R5. Since entities from any CEC belong to more than one CEC, a global identifier should be employed. The identifier employed by Mobile IP is an IP address. An IP address is globally unique. It can be the case, that part of the IP address identifies the MEC (i.e. a home network) and another part of identifies the Mobile IP node itself, within the MEC. Whatever the case is, the identifier for Mobile IP nodes is globally unique, and therefore, this requirement is satisfied in the Mobile IP architecture.

We have now checked all the information level requirements and we have concluded that they are satisfied in Mobile IP. This applies regardless of whether Mobile IPv4 with route optimisation [PJ00] is used or Mobile IPv6 [RFC1883].

5.3.3 Computational viewpoint requirements

From the computational viewpoint, we can see that the Mobile IP architecture employs a Mobile Entity Agent (MEA) design pattern. We can now show that most of the disadvantages of the MEA design pattern, which we identified in section 4.2.2, are evident in the Mobile IP architecture.

Firstly, when datagrams are sent to a Mobile IP node that is away from home, the Home Agent has to be alerted and pick them up on behalf of the Mobile IP node. This adds a processing overhead. Also, there is another processing overhead since datagrams to a Mobile IP node that is away from home have to be routed through the Home Agent, and the Foreign Agent. Triangular routing is a drawback of the Mobile IP architecture when IPv4 is used.

However, this is not the case when IPv6 [RFC1883] or Mobile IPv4 with route optimisation [PJ00] is used. This comes at the cost having to communicate binding information to every node that contacts the mobile node for the first time at a new location.

The advantage that the HA is always available even when a Mobile IP node is detached from the network applies in this case, although a Mobile IP HA cannot cache datagrams on behalf of a Mobile IP node and the protocol does not support this feature. However, this feature is supported by the suggested route optimisation extensions to Mobile IPv4 [PJ00]. There are no call handling options in Mobile IP. For example, there is no standard way for the HA to maintain a list of IP nodes that are not allowed to contact a Mobile IP node when away from home. There is an authentication exchange between a Mobile IP node and its HA.

The fact that a Mobile IP node only has to update its location with one other entity (the HA) when away from home, is an advantage. In Mobile IPv6 however, in order to overcome the triangular routing problem, a Mobile IP node updates its location with more than one IP node that may call it. It could be more efficient if a mobile node in IPv6 only had to update its address with one other entity only (this could be the HA or a special directory service). A scheme where the updated binding information can be communicated to the HA only is supported by the route optimisation extensions to Mobile IPv4 [PJ00].

Similarly, the advantage that the location of a Mobile IP node is unknown outside its HA is true in Mobile IP with IPv4. In Mobile IPv6 however, the location of an IP node can be known outside its HA. The same applies when the route optimisation extensions to Mobile IPv4 are used.

We believe that a combination of the MEA with the Mobile Entity Locator (MEL) design pattern would benefit the Mobile IP architecture. If the HA could be mobile, the triangular routing problem would be solved. Alternatively, the HA could return the new location of the Mobile IP node to the calling IP nodes, so that they can contact the HA directly at its new location. If the first solution were to be applied, the DNS could be used as the Mobile Entity Agent Locator of the MEL-MEA combination. The DNS is slow in updating its records, but on the other hand we can anticipate that the HA will not change location too frequently. If the HA could move closer to the Mobile IP node so that the triangular routing problem is eased, then we could keep the location of a Mobile IP node secured within the HA. The alternative solution has also been followed in Mobile IPv6 implementations [FS98].

Also, Mobile IP could be improved by allowing call handling for calls via the HA and by allowing the HA to cache datagrams while the Mobile IP node is detached from the network

5.4 GSM

Identify the mobile entities in the GSM architecture could not have been straightforward. This is because, as explained in section 2.2, GSM uses the term "Mobile Station" (MS) to identify a coupling between the subscriber of a GSM service (the person) and the mobile equipment that is used to access the GSM service (the mobile terminal).

5.4.1 Mapping the FoT to the GSM architecture

The ETSI standard GSM 02.17 [GSM02.17] allows for a subscriber to insert a Subscriber Identity Module (SIM) card to the mobile terminal equipment that they want to use. This could be seen as two entities in our framework: one entity could be the subscriber (the person) that is identified by the IMSI number, and the other entity could the mobile terminal equipment that is identified by the IMEI number [GSM02.16]. The subscriber can be seen as a mobile entity, since it can be the source of the sink of message exchanges (calls) via the terminal equipment used. However, the mobile terminal equipment can not be seen as a mobile entity according to the FoT because it cannot be the source or the sink of any information exchange. The mobile terminal equipment only conveys information from the mobile subscriber to the Mobile-services Switching Centres (MSC) [GSM03.02]. The IMEI number mainly identifies the capabilities of the mobile terminal equipment and its manufacturer. The IMEI is not used to address any exchange to mobile equipment individually. Therefore, we have only one mobile entity, the Mobile Subscriber (MSub) of a GSM service. Since the terminal equipment is only the conveyor of information, it can be seen as the medium via which a MSub attaches itself to the network.

For our analysis, we employ a communications network in the GSM communications environment that features our mobile network entity, the MSub. We also assume that mobile terminal equipment attaches a MSub to a MSC, the entity which performs location registration procedures for a mobile station and which can route calls through to the mobile station. In our view of the communications network we omit other entities that exist between the MSC and the mobile station, like the Base Station Controller (BSC) and the Base Transceiver Station (BTS) [GSM03.02], in order to keep our framework simple. These entities are not important for our analysis and we can consider them as a part of the MSC in this example. The network entities of our communications network are:

E1. *Mobile Subscriber (MSub).* This is the person that is subscribed to the GSM service and is identified by the International Mobile Subscriber Identity (IMSI) number. The IMSI number together with other subscriber-related information is stored in the SIM card. A MSub can attach itself to different mobile terminal equipment by inserting the SIM card to it. A MSub can be reached via the terminal equipment used. *The Mobile Terminal Equipment (MTE) or mobile equipment* is the medium that a subscriber can use by inserting a SIM card to it. With the MTE, a MSub can attach to different parts of a network served by different MSCs. Therefore, a MSC can be seen as the location of a MSub when it is contacted by other MSubs. Each MSub is subscribed to one Public Land Mobile Network (PLMN). Each PLMN owns and controls a number of MSCs.

E2. *Mobile Station (MS).* This is the coupling between Mobile Equipment and a Mobile Subscriber. As the ETSI GSM standard 02.17 [GSM02.17] specifies, this is a piece of MTE with a SIM attached to it. We do not use this entity in our analysis. We separate the MSub (which is a mobile network entity) from the MTE (which is a medium).

E3. *Mobile-services Switching Centre (MSC).* This is an exchange that performs switching and signalling for the mobile stations that are located in the area that it serves. It also performs location registration for mobile stations. We consider MSCs fixed network entities in our analysis.

E4. *Gateway MSC (GMSC).* This is a MSC that performs the routing function for calls to a mobile station. In order to do that, it interrogates the Home Location Register (HLR) (see section 2.2) to find the location of a mobile station; it then routes the call to the mobile station via the MSC that the mobile station is attached to. We consider GMSCs fixed network entities in our analysis.

E5. *Fixed Station (FS).* This, physically, is terminal equipment of the traditional Public Switched Telephone Network (PSTN). It is identified by the telephone number that can be used to route calls to the FS. The concept of a FS is based on an association between the terminal and the fixed telephony subscriber that uses it. That is because in traditional telephony, the equipment is associated with a person (who can receive telephone calls on this number and who is billed for this service). FSs are fixed network entities. They can have exchanges with other FSs or MSs.

E6. *Local Exchange (LE).* This is the local exchange equipment that is used in the traditional PSTN. FSs are attached to the PSTN through a LE so that they can participate in message exchanges (calls).

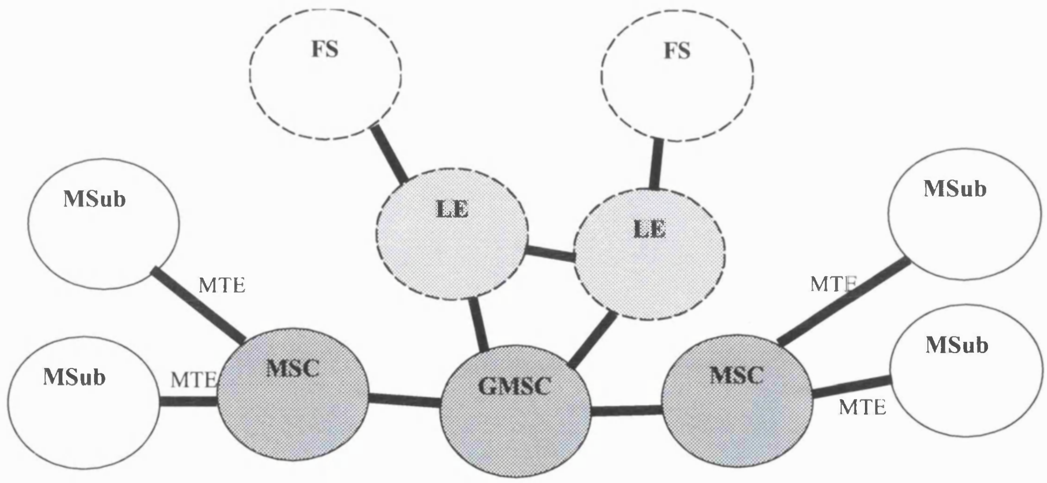


Figure 5-15: Example of a GSM communications network.

These network entities are interconnected via unicast media. We consider the wireless links of MSs to the MSCs unicast media. This is because when the MSC addresses a

MSub via the wireless medium, the MSub has to be addressed individually for the message to go through.

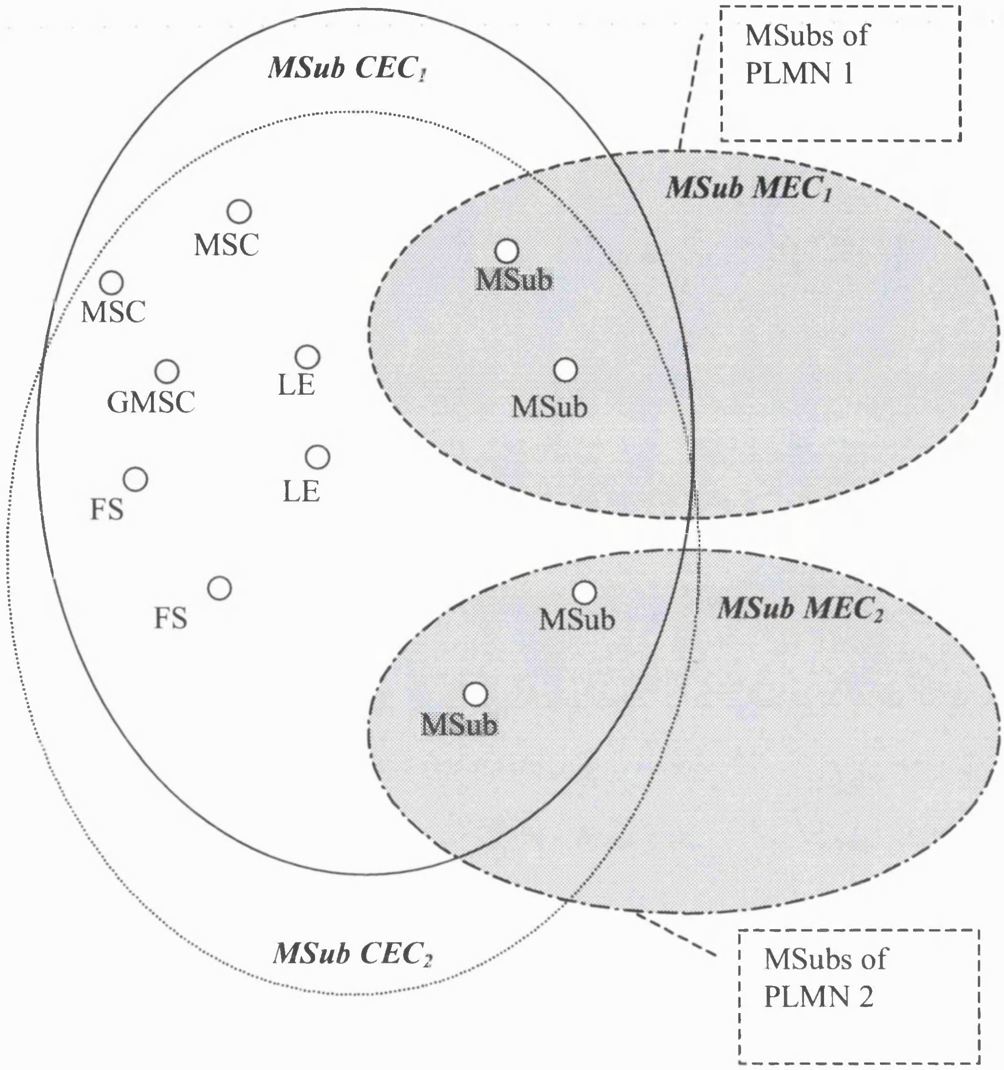


Figure 5-16: Example of MECs and corresponding CECs in a GSM communications network.

We can now distinguish a number of Mobile Entity Classes (MEC) in this communications network. We consider one MSub MEC for each Public Land Mobile Network (PLMN).

MEC *Mobile Subscribers MEC (MSub MEC)*. This comprises all the MSubs of a PLMN that support the GSM service. Therefore, we consider one MSub MEC per PLMN. The Absolute Location Space (ALS) of a MSub MEC is the set of the MSCs in the communications network that a MSub is allowed to roam in. It could be the case that every MSub of any Public Land Mobile Network (PLMN) can roam freely to any

MSC of any PLMN. But this is not necessarily the rule. It can be the case that some PLMNs do not allow their registered MSubs to roam in any PLMN. For this reason, we consider separate MSub MECs, one per PLMN. The Network Entity Class (NEC) that corresponds to each MSub MEC includes all the MSubs, FSs, MSCs, GMSCs and LEs of the GSM communications network. Since the ALS is a subset of the NEC in this case, the Complete Entity Class (CEC) that corresponds to the MSub MEC is identical to the NEC. We have made the assumption that any FS or MSub of any network can call any MSub of any PLMN.

A view of the network entities of a GSM communications network is given in Figure 5-15. An example of two PLMNs with different MSubs subscribed to each of them is given in Figure 5-16; we illustrate the different MSub MECs and their corresponding CECs. For this diagram, we made the assumption that the MSubs of each PLMN are allowed to roam in both PLMNs.

5.4.2 Information viewpoint requirements

In Table 4-1 we summarise the information level requirements for mobile entities in the FoT. Here, we examine each requirement separately to see if they are satisfied in the design of the GSM architecture.

R1. Identifiers are used for Mobile Subscribers (MSub). The International Mobile Subscriber Identity (IMSI) numbers, the Mobile Station ISDN (MSISDN) numbers and the Mobile Station Routing Numbers (MSRN) are used to identify the MSubs to different network entities. Therefore, this first requirement of the FoT is satisfied in the GSM architecture.

R2. The IMSI numbers that are used to identify MSubs are unique in each PLMN, and therefore they are unique in each MSub MEC. The MSISDN numbers are also unique in each PLMN and therefore they are unique in each MSub MEC. The MSRN numbers are used to MSubs are unique at any given time in the MSub MEC. For this reason, this requirement of the FoT is satisfied too.

R3. When a call is to be arranged from one FS or MSub to another MSub, a number of steps will take place. First, the MSub or FS will use the MSISDN number to address the called MSub. The LE or MSC that will receive the request may need to interrogate the Home Location Register (see section 2.2) directly or via a GMSC in order to obtain a corresponding Mobile Station Routing Number (MSRN) [GSM03.04, GSM03.03]. When the MSRN is obtained, the call can be directed to the MSC to which the MSub is attached. The MSC will use the MSRN to pass the call to the MSub. In order for a MSRN to be allocated to a MSub, the IMSI number will have been used when the MSub registers via the MSC. Therefore, we can see that in order for a message exchange to take place, an IMSI number is mapped to a MSRN number and then a MSISDN number is mapped to a MSRN number. Also, we see that a mapping of a MSISDN number to an IMSI number exists. Therefore, the third requirement of the FoT is not satisfied in GSM. The consequence of this is not that the system is not mobile, as explained in section 4.1. The consequence is that the system is more complex because this mapping of identifiers has to take place and because some network entities (like the MSC and GMSC in our example) have to know (or be able to obtain) extra unique identifiers for a mobile entity to convey a message exchange (call).

R4. As we can see from the analysis so far, the CECs that we have for each PSTN have network entities in common. The LEs are an example of network entities shared by more than one CECs. For this reason, the fourth FoT requirement does not apply for this GSM communications network.

R5. All the three identifiers that we described, i.e. IMSI, MSISDN and MSRN, consist of several fields. One field of each identifier identifies a MSub uniquely in a GSM PLMN. These are the MSIN field in the IMSI, the SN field in the MSISDN and the SN field in MSRN [GSM03.03]. Therefore, a part of each identifier identifies a MSub uniquely in the MSub MEC. Other fields identify the GSM PLMN uniquely in the GSM network. These are the MCC and MNC fields in the IMSI, the CC and NDC fields in the MSISDN and the CC and NDC fields in the MSRN. Therefore, a part of each identifier identifies a MSub MEC uniquely in the communications network. We conclude that this requirement of the FoT is satisfied in the GSM architecture.

We have now checked all the information level requirements and we have concluded that only the requirement (R3) is not satisfied in the GSM architecture. We believe that GSM fails to satisfy this requirement because of the legacy of traditional telephony and the strong requirement to make GSM interwork with the traditional telephony.

From this analysis, we saw that the mobile terminal equipment cannot be the source or sink of information exchanges and that the mobile entities of a GSM system are the subscribers of the GSM service and not their mobile terminal equipment. We can now justify the fact that GSM cannot accommodate more than one subscribers served by the same terminal equipment at a given time. The reason is that the terminal equipment is strongly associated with one subscriber at any given time and the terminal equipment cannot be addressed independent of the subscriber that is using it.

5.4.3 Computational viewpoint requirements

We can see that a Mobile Entity Agent (MEA) design pattern is employed in this GSM communications network. There is a Home Location Register (HLR) entity at the home PLMN of mobile subscriber (MSub) that keeps track of the location of the MSub and some personalisation data regarding the MSub [GSM03.02, GSM02.17, GSM03.04]. It is recommended that the HLR is interrogated only by the GMSC of the Home PLMN of a MSub [GSM03.04].

If that is the case, we can consider a Mobile Entity Agent that comprises the HLR and the GMSC of the home PLMN. In this case, we see that the benefits and the consequences of the MEA design pattern are evident in GSM. Starting with the benefits, we see that the HA is available even when the MSub is not attached to the network. The agent could cache or handle call requests if the MSub is unavailable and the MSub has to update its location only with the HLR every time it changes location. Also, we know that the location of a MSub can be concealed from entities outside the home PLMN. The consequences of employing the MEA design pattern in GSM are:

C1. There is a processing overhead, a delay, when a call is to be routed to a roaming MSub. This delay is what it takes to go to the Home PLMN GMSC of a MSub and for this GMSC to obtain the location of the MSub from the HLR.

C2. Since the call is routed via the home GMSC there is an extra overhead to set up the link between the home PLMN and the visiting PLMN of a roaming MSub.

C3. Triangular routing is a disadvantage of the GSM architecture. When a MSub is roaming to a visiting PLMN, the calls to this MSub have to be routed via the GMSC of the Home GMSC of the MSub. This is the recommended approach in the ETSI GSM standard 03.04 [GSM03.04].

C4. Before a MSub is contacted, the home GMSC and the HLR always have to be contacted first. This adds an extra stage to contacting a MSub.

C5. If the HLR or home GMSC of a MSub becomes unavailable, the MSub cannot be contacted although it can be available.

If, despite the recommendation of the ETSI GSM standard 03.04, the originating LE can contact the HLR of a called MSub directly, we have the benefits and the consequences of the variation of the MEA design pattern as presented in section 4.2.2. The benefits are that there is no risk of triangular routing and that the overhead of passing calls to a MSub via its home GMSC is removed. On the other hand, there are come consequences. Firstly, the location of a MSub cannot be concealed from entities outside the home PLMN. Also, the location of a MSub may change during the period between the retrieval of its old location from the HA by the originating LE and the set up of the connection from the originating LE. According to the ETSI GSM 03.04 standard, this approach would also make charging more difficult.

Applying a combination of the MEL and the MEA design patterns to the GSM architecture would have a number of benefits. The HLR of a MSub could migrate to the visiting PLMN where the MSub is roaming. A HLR Locator (HLRL) could point to the location of the HLR at the home or the visiting PLMN where the HLR resides at a given time. In this way, the location of a MSub would still be concealed from entities outside the visiting PLMN. Also, there would be no triangular routing problem since the visiting GMSC that routes calls to the visiting MSub and the MSub will be on the same PLMN at any time. The HLRL could be a distributed database with cached copies of HLR locations (like the DNS in the Internet) so that a HLR location can be obtained from more than one place. Since a MSub is not expected to change PLMN very frequently, the HLR should be a low speed mobile entity. This would help to keep the records of a distributed HLRL consistent at all times.

5.5 The TINA Architecture

In the analysis of the Prospect environment and discussion of the personal mobility support in Prospect, we presented many of the principles of the TINA architecture. TINA employs a combination of the MEL and the MEA design patterns for locating User Agents (UA) and the Persons that the UAs delegate for. Also, call handling can be enabled at the UA level. The Prospect environment is a TINA-like environment [LTMR97, Tirop98] where most of principles of the TINA service architecture [TINA-SA5, TINA-SAA5] are followed.

However, there are a number of requirements to enable TINA to accommodate more kinds of mobility, like Role Mobility.

From the information viewpoint, TINA should respect the requirement (R5) from Table 4-1. It should be recognised that users (persons), User Agents (UA) and other components of a TINA environment may need to have exchanges with members other mobile entities that belong to other MECs, like a Role MEC. Therefore, TINA should support globally unique identifiers that contain a MEC identifier on top of the entity identifier. This could be very much like the "calledPartyIdentifier" that was employed in Prospect [Tirop98, TMK99].

From the computational viewpoint, a migration of the agents to the domain of the mobile entities that they delegate for would offer advantages. For example, exchanges between users and UAs would be more efficient if the UAs move closer to the user.

TINA is a high-level architecture for Telecom services. It is up to the groups that deploy TINA systems to take these extra requirements into account. However, some requirements that we investigated on the FoT could be included in the TINA specifications. Such requirements have already been suggested for enabling role mobility in a TINA environment [Tirop98].

6. Discussion

In the previous chapters we describe the problem area that we are tackling in this thesis. In chapter one and chapter two we justify that a generic approach to the problem of mobility is missing today. In chapter one we describe the different approaches that the standard bodies employ to define and support mobility. In chapter two we discuss the limited scope and the drawbacks of two particular architectures: the Mobile IP architecture [RFC2002] and the GSM architecture [GSM03.02] and we relate our work to other research activities.

In chapter three we suggest a Framework of Terms (FoT) that aspires to provide a terminology for mobile systems in a generic and technology independent way. In the FoT we give a definition of a communications network which features fixed and mobile network entities interconnected by unicast or broadcast media. In chapter four we investigate the requirements for mobility support in the FoT from two different ODP [ISO-ODP1] viewpoints: the information viewpoint and the computational viewpoint. We identify requirements regarding the necessity for mobile entity identifiers and the uniqueness of those identifiers. We employ a design pattern template [MM97] to describe computational entities that are required for locating mobile entities and for delegating for mobile entities while they are unreachable.

The hypothesis of this thesis, as presented in chapter one, is that a framework of terms that can describe a mobile system in a generic way is possible, and that certain requirements for mobility can be expressed in this framework. Also, that this framework can be expressive enough to represent a variety of different mobile systems and, at the same time is of practical value. Finally, we also made the hypothesis that this framework can provide for systems that avoid common problems of existing mobile system architectures.

We planned a particular approach to validate the hypothesis. Firstly, we provided a Framework of Terms (FoT), which was described in chapter three. In order to test the generality of our framework we mapped it to a number of existing mobile systems. In chapter five we provide a mapping of the FoT to the Mobile IP architecture and to the GSM architecture. In chapter four we examined the different requirements on the FoT from the information and computational ODP viewpoints to enhance the practical value of the FoT. In order to investigate its practical value we used the FoT for the deployment of a new system for the management of role mobility in the European ACTS project Prospect as described in chapter five. The implementation of the role mobility component is described in section 5.2. Also, we tested if some well-known problems of existing mobile system architectures can be detected and explained by using the FoT to look at these architectures. This allowed us not only to further investigate the practical value of the FoT but also to demonstrate that the FoT can provide for systems that avoid common problems of existing mobile system architectures.

In this chapter we discuss this approach step by step and we see how the results of our approach relate to investigating the validity of our initial hypothesis.

6.1 Definition of the FoT

The definition of the Framework of Terms (FoT) of chapter three was intended to map to a variety of systems that support mobility. Our aim was to define the FoT at a generic and technology independent level. At this level we defined concepts like communications network, network entities, media, mobility, mobile network entities, fixed network entities and identifiers for network entities.

The terms FoT are defined informally. The informal nature of the FoT definitions provides the advantage that it allows it to be understood by mobile systems designers who are not necessarily familiar with formal models and formal methods. In this way, the experience of these mobile systems designers can be expressed and communicated in FoT terms. It was one of the goals of this thesis to provide a platform for the exchange of ideas among people from different communities and with different backgrounds. On the other hand the informal nature of the FoT has as a consequence a certain amount of ambiguity. For this reason, the FoT concepts are expressed in π-calculus terms in chapter three, in order to reduce their ambiguity.

The FoT is also generic and technology independent. This property of the FoT provides certain advantages. First of all, we can hope for a generic and technology independent definition of mobility, which is something that is missing today, as explained in chapter one. Apart from this, any requirements in a generic framework of terms should be generic enough to apply to any real system that this framework can map to.

On the other hand, there are some consequences of this approach. Since the FoT is on a high level of abstraction, it is possible that some aspects of mobile systems could not be investigated in depth. In particular, issues related to performance, security, fault management or quality of service could not be addressed to a large extent. One of the reasons for this is that the FoT is technology independent, while such issues usually arise with the use of specific technologies.

However, we consider that a generic framework of terms for mobile systems and a definition of mobility in this framework are the first steps that are required towards providing a common infrastructure for the exchange of ideas between the various research communities on mobile systems. As we point out in the introduction of this thesis, this was one of our goals. Having this infrastructure in place, the FoT can be extended and address further areas in the future.

The FoT is also accompanied with some work that relates it to abstract and mathematical modes for mobile systems, although the FoT in itself is not a model but a framework of terms enhanced with a number of requirements descriptive of mobile systems. This was intended to make the FoT approachable by people from a theoretical background and to allow for future theoretical work on solutions expressed in FoT terms.

In the FoT we attempted a grouping of mobile network entities in Mobile Entity Classes (MEC). The members of the MEC are the mobile entities that can share the same locations. This is not necessarily the only grouping that we can apply to the mobile entities of the FoT. However, we consider this grouping particularly important for two reasons. First of all because it shows the requirement that if a number of mobile entities can share the same locations then, when contacting a particular entity (via its location), we should be able to distinguish this entity from the others by means of a unique identifier. Second, this grouping of mobile entities can be identical to other groupings of mobile entities that already exist in mobile systems today, like groupings according to their physical characteristics; this makes mapping the FoT to real systems easier. For example, in the Prospect environment the *Persons MEC* is the group of the mobile entities that share the *terminals* of the Prospect system as their location space. On the other hand, the *Persons MEC* also describes the group of the mobile entities that share the physical characteristics of a *person* or *user*.

In the FoT we specify a Network Entity Class (NEC) as the set of network entities that perceive the members of a MEC at the same location at any given time. This grouping is very useful. In the FoT we specify that the location of each mobile entity depends on a particular message exchange to it that originates from a particular network entity; for this reason we call the location of mobile entity a *relative location*. By employing NECs, we can see the locations of the members of a MEC not relative but absolute, to the members of the corresponding NEC. This simplifies the specification of requirements on mobile entity identities in the FoT.

6.2 Requirements on the FoT

As we pointed out in the introduction of this thesis, we did not intend to provide a prescriptive framework but a descriptive one. Therefore, we do not provide a methodology on how the FoT can map to real systems. Similarly, we do not provide a methodology on how the requirements of the FoT can be applied to the design of a real system.

We used the ODP viewpoints [ISO-ODP1] to investigate the requirements for mobility in the FoT. We chose the ODP viewpoints because it is believed that they can cover any aspect of a distributed system [FLM95]. In section 1.4.2.2 we justified why a mobile system is by nature distributed.

We investigated two different ODP viewpoints: the information viewpoint and the computational viewpoint. From the information viewpoint we specified a number of requirements for mobility on the FoT. We explored the necessity of identifiers for mobile entities, the uniqueness of those identifiers and their scope of uniqueness. From the computational viewpoint we investigated the requirements for locating mobile entities and for delegating for mobile entities while they are not attached to the communications network. We presented possible solutions to meet these requirements in the form of design patterns [GHJV97] using a CORBA design pattern template [MM97].

However, there is no claim that the two design patterns can provide for better or optimal mobile systems. Also, there is no claim that the two design patterns are axiomatic in any sense. The two problem areas that these two design patterns try to solve are considered axiomatic, as explained in section 4.2. MEL and MEA as *analysis design patterns* provided an insight into pros and cons of mobile system design in Mobile IP, GSM and Prospect.

We cannot claim that every information or computational aspect of a mobile system has been explored in this thesis. But we do believe that the information viewpoint and the computational viewpoint of the FoT have been explored to an extent that justifies the practical value of the FoT. In section 6.1 we explain the reasons that determined the scope of the FoT.

6.3 Mapping the FoT to real systems

In chapter five we showed how the FoT mapped to the Prospect environment and how it maps to the Mobile IP and the GSM architectures.

The first step towards mapping the FoT to a real system was to decide the mapping of the communications network concept of the FoT to different aspects of the real system. As explained in chapter three, a communications network is a particular view of a communications environment from a specific viewpoint. In order to map the FoT to a real system, we should map the concept of a communications network to every aspect of the real system.

For the Prospect environment, as described in section 5.1, we specified the view of the component designer for role mobility and personal mobility support. In that model we covered all the mobile entities in Prospect (persons and roles) and the computational entities of the Prospect architecture. In section 5.2 we present the implementation of that component.

For the Mobile IP architecture of section 5.3, and for the GSM architecture of section 5.4, we chose similar views that featured the mobile entities of these architectures. With this mapping we were able to tell exactly which kinds of mobility Mobile IP and GSM can support.

From the experience of mapping the FoT to these real systems we can see that once the concept of the communications network maps to a view of the real system, the mapping of mobile entities, MECs, NECs and CECs can follow easily. From the methodology point of view, deciding the view of the real system on which the communications network will map is an ambiguous part. It was not intended to provide a methodology for this mapping in this thesis. However, if a system designer did all the possible mappings of the communications network to their real system, they could then arrange for the requirements of the FoT to be respected on all these aspects so that their system could support mobility efficiently.

By providing these mappings of the FoT to real systems that come from different communities with different cultures (such as the Internet and the Telecom communities) we have supported our claim that the FoT is generic enough but also expressive enough to map to a variety or real systems.

6.4 Mapping the FoT requirements to real systems

In chapter five we showed how the FoT requirements were respected in the design of the Prospect system. We also showed how the role mobility system was implemented and demonstrated successfully during the last trial of Prospect.

Relating this fact to our hypothesis, we showed how the FoT mapped to a real system and how the requirements of the FoT were respected in the design of the real system. Then we showed how this real system was implemented and how it was demonstrated to support mobility. In this way, we investigated the validity of our claim that the FoT is of practical value.

Also, in chapter five we showed how the FoT requirements were not respected in some parts of the Mobile IP and the GSM architectures and we demonstrated how some well-known problems of these architectures could be avoided by respecting these requirements. In this way, we have confirmed the practical value of the FoT further. Also, this shows that the FoT can aid designers in considering all aspects of mobility and thus lead to coherent solutions which avoid pitfalls common in today's systems.

6.5 Drawbacks

Through the process of verifying our hypothesis we were able to identify some drawbacks of the FoT and the FoT requirements. These drawbacks we discuss in this section.

D1. The FoT introduces a new terminology for mobile systems. It may take some effort to familiarise oneself with this new terminology. Some terms like media, unicast and broadcast have different meanings in other existing standards or architectures.

D2. The FoT is not a prescriptive model. Because of this, it is not straightforward how the concept of the communications network in the FoT terms can map to real systems. A prescriptive approach would assist towards designing mobile systems and benefiting from the FoT and the FoT requirements that we provide. In this way, a prescriptive model would further enhance the practical value of the FoT.

D3. We do not provide a methodology on how existing systems that do not respect some of the FoT requirements could encompass these requirements in their existing design. This is out of the scope of our thesis since the FoT was not intended to be a prescriptive model.

D4. The FoT does not provide any guidelines or design requirements from the ODP enterprise viewpoint. As stated in the introduction of this thesis, it was considered that enterprise viewpoint requirements would be practically hard to verify. Our intention was to provide a framework that is generic enough to map to a variety of mobile systems and at the same time of practical value. The practical value of the FoT is confirmed to our satisfaction by addressing the information and computational viewpoints. We believe however, that an investigation of requirements from the enterprise viewpoint would provide valuable further work.

D5. The engineering viewpoint has not been investigated in the FoT. Although some engineering viewpoint issues were discussed together with the computational level issues in section 4.2, a detailed investigation would be helpful for mobile systems design. In order to do that, the FoT concept of a medium should be enhanced with attributes related to performance, security, or fault tolerance. Similarly, the FoT concept of a mobile network entity should be enhanced with similar attributes including energy reservoir; an attribute that in real systems applies to mobile terminals, which can have limited battery power. It was estimated that investigating engineering viewpoint considerations involves a large amount of work and resources. Furthermore, they would require a wide discussion and a consensus from different communities with different cultures in order to safeguard the generality of the FoT. We believe that the current work on the FoT provides a good platform for this purpose.

D6. Some technologies could be evaluated in the FoT and some of their benefits or drawbacks could be identified. In this way, the technology ODP viewpoint could be addressed in the FoT. However, this area is out of the scope of this thesis.

We have now discussed the results of our research and experimentation and we have shown how the validity of our hypothesis was investigated through our approach. We have also identified the benefits and the drawbacks of our solution.

In the next chapter we present our conclusions and some issues that we are considering for further study.

7. Conclusions and further work

Based on the discussion of the previous chapter, here we present our conclusions regarding the verification of our hypothesis and further work that can be undertaken to advance our research.

7.1 Conclusions

A first conclusion, which can be established from the analysis in chapter one and in chapter two is that a common definition of mobility is missing today. The different research and standards communities like the Internet and the Telecom communities use different terminology and adopt parochial views on mobility.

In this thesis, with the definition of the Framework of Terms (FoT) we have provided a generic definition of a communications network, which can map to diverse communication environments. On this generic framework we were able to provide a definition of mobility.

Apart from this, we investigated the requirements for managing mobile entities in this framework and we provided a number of requirements from the information ODP viewpoint and the computational ODP viewpoint for mobility support in the FoT. From the discussion in chapter six we can conclude that the FoT is generic enough to map to a variety of real systems.

We can also conclude that the FoT is of practical value. This can be justified from the fact that a real system, the role mobility component in the European ACTS project Prospect, was implemented respecting the FoT requirements and it was demonstrated that it supports mobility efficiently. Additional to this, examining the Mobile IP architecture and the GSM architecture through the FoT allowed us to explain well known problems of these architectures and to see how they could be avoided.

The FoT provides a framework for an exchange of ideas on mobility between different communities. We can justify that with the fact that the FoT is generic enough to map to different mobile systems architectures from the Telecom community (like the GSM architecture) and the Internet community (like the Mobile IP architecture). Because of its informal nature, the FoT terms can be accessible to mobile systems designers who are not necessarily familiar with formal methods and formal models.

Apart from this, the FoT can be related to existing abstract and mathematical models for mobility. This could allow conveying ideas that have been expressed formally to a larger number of diverse communities. Using π-calculus notation to express FoT terms the ambiguity of the FoT definitions was reduced.

With the FoT we were able to investigate novel areas of mobility, such as role mobility in the Prospect project. In Prospect, we were able to model the *virtual* mobile entity of a role and we designed a component for role mobility in line with the requirements of the FoT. This component was implemented and demonstrated.

We found that the concept of a Mobile Entity Class (MEC) and the corresponding Network Entity Class (NEC) and Complete Entity Class (CEC) of the FoT (see chapter three) could map to a number of diverse environments. We found the MEC concept very powerful since it allowed us to investigate the requirements for unique identifiers and the scope of uniqueness for those identifiers.

Additionally, the design pattern approach allowed us to investigate FoT requirements from the computational ODP viewpoint. We found that the design patterns for a Mobile Entity Locator (MEL) and a Mobile Entity Agent (MEA) could map to a variety of real environments such as Prospect, Mobile IP and GSM.

Summary of thesis contribution   **C1.** We provided a Framework of Terms (FoT) which is generic and technology independent to represent a variety of existing mobile systems. This was verified by mapping the FoT to different mobile systems such as Mobile IP and GSM. In order to disambiguate the concepts of the FoT that were given informally, FoT terms are also expressed using a formal notation (π-calculus notation).   **C2.** In the FoT we provided a generic and technology independent definition of mobility.   **C3.** In the FoT we investigated the requirements for mobility from two different ODP viewpoints: the information viewpoint and the computational viewpoint. From the information viewpoint we investigated the requirements regarding the identification of mobile entities. From the computational viewpoint we provided a number of Design Patterns for locating or delegating for mobile entities.   **C4.** The FoT is of practical value. It was mapped to existing mobile system architectures such as Mobile IP and GSM and it explained some well-known problems of these architectures; it showed which of the FoT requirements these architectures fail to satisfy. Also, it allowed us to suggest how these problems could have been avoided. This also demonstrates that the FoT can aid designers in considering many aspects of mobility and thus lead to coherent solutions which avoid pitfalls common in today's systems.   **C5.** The practical value of the FoT is also supported by designing, implementing and demonstrating a mobile system, which respects the FoT requirements. The role mobility component was successfully implemented and demonstrated by the European ACTS project Prospect.

Table 7-1: Thesis contribution.

Also, the suggested combination of the MEL and MEA design patterns of chapter four assisted in the optimisation of the role mobility component in Prospect and provided possible solutions for the problems of the GSM and the Mobile IP architectures. We found that the MEL and the MEA design patterns provide a very good tool for identifying the pros and cons of a mobile system from the early stages of the design process.

The concept of a communications network in the FoT allowed us to investigate mobility in real systems at different levels. It is very important to decide how to map the concept of the communications network to a real system, since there are many views on a real system that the communications network concept of the FoT can map to. Ideally, the communications network concept must be mapped to the real system in every possible way, and the mobility requirements of the FoT must be investigated for every mapping.

A summary of the contribution of this thesis is provided in Table 7-1.

7.1.1 Experiences from the design of the role mobility component in Prospect

The design of role mobility in Prospect was a challenging exercise. It allowed us to investigate a novel area of mobility and to respect the FoT requirements in it. In this section we present our conclusions regarding this exercise.

From the observation that many exchanges in communications services are role-based and from the plethora of the service roles that were identified for a number of services in Prospect (although a few were implemented), we can conclude that role management is an essential part of service management. The advanced services that we expect to see widely available in the near future (such as multimedia conferencing, Tele-training, and others) require a significant number of service independent roles. Also, many of these services require service specific roles (such as the "tutor" role for the Tele-education service) which can be specified and deployed in a generic and efficient way. Requirements for reusability and modularity of service management functionality led us to the deployment of a generic, reusable service management component, which could be integrated in the TINA-like environment of Prospect next to the subscription and accounting management.

There are some requirements though, on the service control plane too. Role-based exchanges must be considered on this level and the service control should allow for service users to be able to address and contact service roles and vice-versa. Also, role-to-role exchanges should also be supported. We only required a minimum number of modifications to the TINA-like service control of Prospect in order to allow for role-based exchanges.

By modelling a service role as a mobile entity and by treating the concept of role mobility in a way similar to other kinds of mobility (such as personal and terminal mobility) we could use the same design patterns that could also apply for supporting personal mobility or terminal mobility. This exercise also gave us an insight into the nature of mobility. A service role as a mobile entity has some peculiarities, which may be encountered on other kinds of mobility too. One peculiarity is that a service role is always mobile. Since a role is carried by a person, and since a person cannot be continuously available, the service role has to move from person to person frequently. Another peculiarity is that a service role can have many instances available at a given time. This is a quality that we do not encounter in personal mobility but which we encounter in agent mobility with multiple agent instances [BM98].

The role management component that was deployed in the Prospect multi-service environment proved generic and useful to the service administrators. Roles could be rapidly defined and deployed. The automated interactions between the role management system and the role-holders (for adding/activating role-holders) were efficient and they saved time and effort for the service administrator. The graphical user interface provided a comprehensible abstraction of the concepts of service roles and role management to the service administrators.

7.2 Further work

Based on this thesis, one of the first priorities for further work would be to enhance the Framework of Terms (FoT) with more attributes. As we discussed in chapter six, enhancing the FoT concepts with more attributes would allow us to investigate the FoT from more ODP viewpoints. For example, adding performance and security attributes to the FoT concepts of a network entity and a medium would allow us to investigate areas such as performance and security management at the generic level of the FoT. This would give us an insight to requirements in the FoT from the engineering viewpoint.

Other ODP viewpoints that could be investigated in the future are the enterprise and the technology viewpoints. From the enterprise viewpoint we could investigate the introduction of new, optional concepts like network entity ownership, medium ownership, service provider, service customer and domain without limiting the generic scope of the FoT. With these new concepts we could investigate enterprise aspects of the FoT and examine the requirements for mobility in the FoT from the enterprise viewpoint.

From the technology ODP viewpoint we could evaluate existing technologies with regard to mobility requirements and point out the pros and cons of each technology in this respect.

An extended FoT that includes requirements from all the ODP viewpoints would allow us to look at a prescriptive model or a methodology for the deployment or optimisation of real systems for mobility support. A prescriptive model would provide a step-by-step guide on how the FoT concept of a communications network would map to real systems and it would indicate the views of a real system that this concept should be mapped to. It would also provide guidelines on how the FoT requirements could be validated for a real system and a methodology on how a mobile system could be deployed.

In particular, the FoT can be described in formal terms such as the π-calculus terms based on the material provided in chapter three. Also, the FoT could be enhanced with the concept of administrative domains by borrowing concepts form the work on the mobile ambients calculus as presented in chapter three. Computational solutions of the FoT could be rigorously tested by employing mobile UNITY.

We are considering further dissemination of our results to standards bodies towards establishing this common language for mobile systems and a common understanding of mobility. Standards groups like the ISO group for ODP and the TINA consortium are on the top of our list.

We have already published some requirements on the TINA Service Architecture for role mobility support [Tirop98]. These requirements apply to both the service management and the service control plane of TINA.

Finally, the FoT could be applied to the design of more mobile systems in the future and it could be used to investigate mobility problems for a larger number of existing mobile systems. This would allow us to deliver further refinements and enhancements to the FoT.

Glossary

A collection of selected acronyms and definitions used in this dissertation.

ACTS	Advanced Communications Technologies & Services
ALS	Absolute Location Space
ANSA	Advanced Network Systems Architecture
BB-ISDN	Broad-Band ISDN
CC	Country Code
CEC	Complete Entity Class
CNT	Communications Network Topology
CO	Computational Object
CORBA	Common Object Request Broker Architecture
CP	Communications Path
CPEP	Communications Path End-Point
CSA	Common Service Architecture
DNS	Domain Name System
DP	Design Pattern
EIR	Equipment Identity Register
ETSI	European Telecommunications Standards Institute

FA	Foreign Agent
FE	Fixed Network Entity
FID	Fixed Entity Identification
FoT	Framework of Terms
FPLMTS	Future Public Land Mobile Telecommunication System
FS	Fixed Station
GID	Globally unique Identifier
GMSC	Gateway Mobile-services Switching Centre
GSM	Global System for Mobile communications
HA	Home Agent
HLR	Home Location Register
ID	Identifier
IETF	Internet Engineering Task Force
IMEI	International Mobile Equipment Identity
IMEISV	IMEI Software Version number
IMSI	International Mobile Station Identifier
IMT-2000	International Mobile Telecommunications 2000
IN	Intelligent Network
IOR	Interoperable Object Reference
ISDN	Integrated Services Digital Network

ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU	International Telecommunication Union
ITU-R	International Telecommunication Union – Radiocommunications standardization sector
ITU-T	International Telecommunication Union – Telecommunications standardization sector
LE	Local Exchange
LS	Location Space
MCC	Mobile Country Code
ME	Mobile Network Entity
MEA	Mobile Entity Agent
MEC	Mobile Entity Class
MEL	Mobile Entity Location
MMC	Multi-Media Conferencing
MMTS	Multi-Media Tele-Service
MNC	Mobile Network Code
MS	Mobile Station
MSC	Mobile-services Switching Centre
MSIN	Mobile Subscriber Identification Number
MSISDN	Mobile Station International ISDN Number

MSRN	Mobile Station Roaming Number
MSub	Mobile Subscriber
MTE	Mobile Terminal Equipment
NB-ISDN	Narrow-Band ISDN
NDC	National Destination Code
NE	Network Entity
NEC	Network Entity Class
NIF	Fixed and Mobile Interworking
ODP	Open Distributed Processing
OMG	Object Management Group
OMT	Object Modelling Technique
OSI	Open Systems Interconnection
OSI-RM	Open Systems Interconnection Reference Model
PLMN	Public Land Mobile Network
PMUAP	Provider Management User Application
PSCS	Personal Service Communication Space
PSTN	Public Switched Telephone Network
RA	Role Agent
RACE	Research for Advanced Communications in Europe
RBAC	Role-Based Access Control

RL	Role Locator
RM-ODP	Open Distributed Processing Reference Model
RR	Role Registrar
SIM	Subscriber Identity Module
SN	Subscriber Number
SP	Service Provider
TA	Terminal Agent
TES	Tele-Education Service
TINA	Telecommunications Information Networking Architecture
TINA SA	TINA Service Architecture
TINA-C	Telecommunications Information Networking Architecture Consortium
TMN	Telecommunications Management Network
UA	User Agent
UAP	User Application
UL	User Locator
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
UPT	Universal Personal Telephony
USCM	Universal Service Component Model
VHE	Virtual Home Environment

VLR	Visitor Location Register

References

[AG97] Martin Abadi, Andrew D. Gordon, "A calculus for cryptographic protocols", Proceedings of the Fourth ACM Conference on Computer and Communications Security, Zurich, Apr 1997.

[ANSA] APM Ltd., "The ANSA Reference Manual Release 1.00", APM Limited, United Kingdom, Mar 1989.

[BFMC97] Ken Buchanan, Rodger Fudge, David McFarlane, Tim Phillips, Akio Sasaki, Howard Xia, "IMT-2000: Service Provider's Perspective", IEEE Personal Communications, p.p. 8-13, Aug 1997.

http://www.comsoc.org/

[BM98] Markus Breugst, Thomas Magedanz, "On the usage of standard mobile agent platforms in telecommunication environments", Proceedings of the 5^{th} international conference on Intelligence in Services and Networks, IS&N '98, Antwerp, Belgium, May 1998.

[Booch94] G. Booch, "Object Oriented Analysis and Design", Addison-Wesley, Reading, MA, USA, 1994.

[BPP96] P. Bhattacharya, B. Patel, C. Perkins, "Preference for Multicast Support with Mobile IP", Internet Draft draft-partha-mobileip-mcastpref-00, Internet Engineering Task Force (IETF), 22 Feb 1996.

[BS98] Gordon Blair, Jean Bernard Stefani, "Open Distributed Processing and Multimedia", Addison Wesley Longman Ltd, England, 1998.

[BT79] B. J. Biddle, E. J. Thomas, "Role theory: concepts and research", Robert E. Krieger Publishing Company, New York, 1979.

[CG97] Luca Cardelli, Andrew D. Gordon, "Mobile Ambients - Annex", Jun 1997.

http://www.luca.demon.co.uk/Bibliography.html

[CG98] Luca Cardelli, Andrew D. Gordon, "Mobile Ambients", Foundations of Software Science and Computation Structures, Lecture Notes in Computer Science 1378, Springer-Verlag, 1998.

[CM88] K. M. Chandy, J. Misra, "Parallel Program Design: A Foundation", Addison-Wesley, 1988.

[CORBAsvc] Object Management Group, "CORBA Services: Common Object Services Specification", Nov 1997.

http://www.omg.org/library/csindx.html

[CORBAv1] Object Management Group, "The Common Object Request Broker Architecture: Architecture and Specification", 91.12.1, 10 Dec 1991.

http://www.omg.org/

[CORBAv2] Object Management Group, "The Common Object Request Broker Architecture: Architecture and Specification, revision 2.2", Feb 1998.

http://www.omg.org/corba/corbaiiop.html

[Davies96] Nigel Davies, "The Impact of Mobility on Distributed Systems Platforms", Proceedings of the 1st International Conference on Distributed Platforms, Dresden, Germany, Feb/Mar 1996.

ftp://ftp.comp.lancs.ac.uk/pub/mpg/MPG-96-15.ps.Z

[DF95] Joao Schwarz DaSilva, Bosco E. Fernandes, "The European Research Program for Advanced Mobile Systems", IEEE Personal Communications, p.p. 14-19, Feb 1995.

http://www.comsoc.org/

[DFWB98] Nigel Davies, Adrian Friday, Stephen Wade, Gordon Blair, "L2imbo: A Distributed Systems Platform for Mobile Computing", ACM Mobile Networks and Applications (MONET), Special Issue on Protocols and Software Paradigms of Mobile Networks, Volume 3, Number 2, p.p. 143-156, Aug 1998.

ftp://ftp.comp.lancs.ac.uk/pub/mpg/MPG-97-03.ps.gz

[Dhawan97] Chander Dhawan, "Mobile Computing: A Systems Integrator's Handbook", McGraw-Hill Series on Computer Communications, The McGraw-Hill Publishing Companies, Inc, United States, 1997.

[DIE97] Joao Schwarz Dasilva, Demosthenes Ikonomou, Heiko Erben, "European R&D Programs on Third-Generation Mobile Communication Systems", IEEE Personal Communications, p.p. 46-52, Feb 1997.

http://www.comsoc.org/

[DNI95] Fabrice Dupuy, Gunnar Nilsson, Yuji Inoue, "The TINA Consortium: Toward Networking Telecommunications Information Services", IEEE Communications Magazine, p.p. 77-83, Nov 1995.

http://www.comsoc.org/

[E.168] ITU-T recommendation E.168, "Application of E.164 Numbering for UPT", Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T), Mar 1993.

http://www.itu.int/

[F.850] ITU-T recommendation F.850, "Principles of Universal Personal Telecommunication (UPT)", Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T), Mar 1993.

http://www.itu.int/

[FHVN98] Gary Fleming, Amre El Hoiydi, Johan De Vriendt, Georgios Nikolaidis, Flavio Piolini, Maria Maraki, "A Flexible Network Architecture for UMTS", IEEE Personal Communications, p.p. 8-15, Apr 1998.

http://www.comsoc.org/

[FLM95] Kazi Farooqui, Luigi Logrippo, Jan de Meer, "The ISO Reference Model for Open Distributed Processing: an introduction", Computer Networks and ISDN Systems 27, p.p. 1215-1229, Elsevier Science B. V., 1995.

[Francis94] Paul Francis, "Addressing in Internetwork Protocols", PhD Thesis, University College London, United Kingdom, Sep 1994.

http://www.ingrid.org/francis/thesis.ps.gz

[FS98] Joe Finney, Andrew Scott, "Implementing Mobile IPv6 for Multimedia", Proceedings of the GEMISIS/IEE/BCS Symposium on Multimedia Network Technology, Digest Number G/MNT/1/1998(e), Salford, United Kingdom, May 1998.

ftp://ftp.comp.lancs.ac.uk/pub/mpg/MPG-98-30.ps.Z

[GC99] Andrew D. Gordon, Luca Cardelli, "Equational Properties of Mobile Ambients", Proceedings of Conference on Foundations of Software Science and Computation Structures, Amsterdam, the Netherlands, Mar 1999.

[GD98] Marie-Pierre Gervais, Alioune Diagne, "Enhancing Telecommunications Service Engineering with Mobile Agent Technology and Formal Methods", IEEE Communications Magazine, Jul 1998.

http://www.comsoc.org/

[GHJV97] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley Publishing Company, Inc, United States, 1997.

[GJS96] James Gosling, Bill Joy, Guy Steel, "The Java Language Specification", Addison-Wesley, Sep 1996.

http://java.sun.com/docs/books/jls/index.html

[GKGF95] C. Görg, S. Kleier, M. Guntermann, M. Frölich, H. Bisseling, "A European Solution for Advanced UPT: Integration of Services for Personal Communication", Proceedings of the Telecommunications Information Networking Architecture Conference, TINA '95, Melbourne, Australia, 1995.

[GSM02.16] GSM 02.16, "Digital cellular telecommunications system; International Mobile station Equipment Identities (IMEI)", European Telecommunications Standards Institute, Nov 1996.

http://www.etsi.fr/

[GSM02.17] GSM 02.17, "Subscriber Identity Modules, Functional Characteristics", European Telecommunications Standards Institute, Feb 1992.

http://www.etsi.fr/

[GSM03.02] GSM 03.02, "Digital cellular telecommunications system (Phase 2+); Network Architecture", European Telecommunications Standards Institute, Mar 1996.

http://www.etsi.fr/

[GSM03.03] GSM 03.03, "Digital cellular telecommunications system (Phase 2+); Numbering, addressing and identification", European Telecommunications Standards Institute, Mar 1996.

http://www.etsi.fr/

[GSM03.04] GSM 03.04, "Digital cellular telecommunications system; Signalling requirements relating to routing of calls to mobile subscribers", European Telecommunications Standards Institute, Nov 1996.

http://www.etsi.fr/

[GSM03.08] GSM 03.08 version 5.1.0, "Digital cellular telecommunications system (Phase 2+); Organization of subscriber data", European Telecommunications Standards Institute, Apr 1996.

http://www.etsi.fr/

[GSM03.12] GSM 03.12, "Digital cellular telecommunications system; Location registration procedures", European Telecommunications Standards Institute, Nov 1996.

http://www.etsi.fr/

[GSM09.02] GSM 09.02, "Digital cellular telecommunications system (Phase 2+); Mobile Application Part (MAP) specification", European Telecommunications Standards Institute, Aug 1996.

http://www.etsi.fr/

[Hamada97] Takeo Hamada, "Dynamic Role Creation from Role Class Hierarchy - Security Management of Service Session in Dynamic Service Environment", Proceedings of the Telecommunications Information Networking Architecture Conference, TINA '99, Santiago, Chile, Nov 1997.

[Hamada98] Takeo Hamada, "Role-Based Access Control in Telecommunication Service Management – Dynamic Role Creation and Management in TINA Service Environment", Proceedings of the Role-Based Access Control Conference, RBAC '98, Fairfax, VA, Oct 1998.

[HC95] Javier Huélamo, Eugenio Carrera, "Distributed Architecture for Advanced PSCS Services in an ATM Network", Proceedings of the Telecommunications Information Networking Architecture Conference, TINA '95, Melbourne, Australia, 1995.

[I.114] ITU-T recommendation I.114, "Glossary of Terms for Universal Personal Telecommunication", Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T), Mar 1993.

http://www.itu.int/

[I.312] CCITT recommendation I.312/Q.1201, "Principles of Intelligent Network Architecture", International Telecommunication Union (ITU), Oct 1992.

http://www.itu.int/

[I.318] CCITT recommendation I.318/Q.1202, "Intelligent Network – Service Plane Architecture", International Telecommunication Union (ITU), Oct 1992.

http://www.itu.int/

[I.373] ITU-T recommendation I.373, "Network Capabilities to Support Universal Personal Telecommunication (UPT)", Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T), Mar 1993.

http://www.itu.int/

[INM95] Ichiro Iida, Takashi Nishigaya, Koso Murakami, "DUET: An Agent-Based Personal Communications Network", IEEE Communications Magazine, p.p. 44-49, Nov 1995.

http://www.comsoc.org/

[ISO-ODP1] ISO/IEC 10746-1 | ITU-T X.901, "Basic Reference Model of Open Distributed Processing - Part 1: Overview", International Organization for Standardization, 1995.

http://www.iso.ch/

[ISO-ODP2] ISO/IEC 10746-2 | ITU-T X.902, "Basic Reference Model of Open Distributed Processing - Part 2: Foundations", International Organization for Standardization, 1995.

http://www.iso.ch/

[ISO-ODP3] ISO/IEC 10746-3 | ITU-T X.903, "Basic Reference Model of Open Distributed Processing - Part 3: Architecture", International Organization for Standardization, 1995.

http://www.iso.ch/

[ISO-ODP4] ISO/IEC 10746-4 | ITU-T X.904, "Basic Reference Model of Open Distributed Processing - Part 3: Architectural Semantics", International Organization for Standardization, 1995.

http://www.iso.ch/

[Kitson95] Barry Kitson, "CORBA and TINA: The Architectural Relationships", Proceedings of the Telecommunications Information Networking Architecture Conference, TINA '95, Melbourne, Australia, 1995.

[KM96] S. Krause, T. Magendanz, "Mobile Service Agents enabling Intelligence on Demand in Telecommunications", Proceedings of the IEEE GLOBCOM '96 conference, 1996.

[KPMF98] Birgit Kreller, Anthony Sang-Bum Park, Jens Meggers, Gunnar Forsgren, Ernö Kovacs, Michael Rosinus, "UMTS: A middleware Architecture and Mobile API Approach", IEEE Personal Communications, p.p. 32-38, Apr 1998.

http://www.comsoc.org/

[LM96] Ulf Leonhardt, Jeff Magee, "Towards a General Location Service for Mobile Environments", Proceedings of the 3rd International workshop on services in Distributed and Networked Environments, Macau, IEEE CS Press, Jun 1996.

ftp://dse.doc.ic.ac.uk/dse-papers/locman/sdne96.ps.gz

[LM98] Ulf Leonhardt, Jeff Magee, "Multi-sensor Location Tracking", Proceedings of the 4th International Conference on Mobile Computing and Networking, Dallas, Texas, Oct 1998.

ftp://dse.doc.ic.ac.uk/dse-papers/locman/mobicom98.ulf_leonhardt.ps.gz

[LMSY95] Emil C. Lupu, Damian A. Marriott, Morris S. Sloman, Nicholas Yialelis, "A policy based role framework for access control", First ACM/NIST Role Based Access Control Workshop, Gaithersburg, USA, Dec 1995.

[LS97] Emil C. Lupu, Morris Sloman, "Towards a role based framework for distributed systems management", Journal of Network and Systems Management, Volume 5, No 1, Plenum Press, p.p. 5-30, Mar 1997.

[LTMR97] D. Lewis, T. Tiropanis, A. McEwan, C. Redmond, V. Wade, R. Bracht, "Inter-domain integration of services and service management", Proceedings of the fourth international conference on intelligence in services and networks, IS&N '97, Cernobbio, Italy, May 1997.

[M.3010] ITU-T recommendation M.3010, "Principles for a Telecommunications Management Network", Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T), Oct 1992.

http://www.itu.int/

[M.687-1] ITU-R recommendation M.687-1, "Future Public Land Mobile Telecommunication Systems (FPLMTS)", Radiocommunication Standardization Sector of the International Telecommunication Union (ITU-R).

http://www.itu.int/

[M.687-2] ITU-R recommendation M.687-2, "International Mobile Telecommunications-2000 (IMT-2000)", Radiocommunication Standardization Sector of the International Telecommunication Union (ITU-R), 1999.

http://www.itu.int/

[Milner89] Robin Milner, "Communication and Concurrency", Prentice Hall, 1989.

[Milner91] Robin Milner, "The Polyadic π-calculus: a Tutorial", Technical Report ECS-LFCS-91-180, LFCS, Dept. of Computer Science, Edinburgh University, October 1991.

[MM97] Thomas J. Mowbray, Raphael C. Malveau, "CORBA Design Patterns", John Wiley & Sons, Inc, United States, 1997.

[MPW92] R. Milner, J. Parrow and D. Walker, "A calculus of Mobile Processes (Parts I and II)", Journal of Information and Computation, Volume 100, Number 1, p.p. 1-40 and 41-77, Sep 1992.

[MR99] Peter J. McCann, Gruia-Catalin Roman, "Modelling Mobile IP in Mobile UNITY", ACM Transactions on Software Engineering and Methodology, p.p. 115-146, Apr 1999.

[NIF-G3] NIF chain guideline number 3, "Service Architecture Aspects in an Integrated Fixed and Mobile Environment", ACTS programme of the European Commission, 1997.

http://www.uk.infowin.org/ACTS/ANALYSYS/CONCERTATION/CHAINS/NI/desc_nif.htm

[OSI-RM] Recommendation X.200 (07/94), "Open Systems Interconnection - Basic reference model: The basic model", Telecommunication Standardization Sector of the International Telecommunication Union, Jul 1994.

http://www.itu.int/

[Pand97] Raj Pandya, "Numbers and Identities for Emerging Wireless/PCS Networks", IEEE Personal Communications, p.p. 8-14, Jun 1997.

http://www.comsoc.org/

[Parrow95] Joachim Parrow, "Interaction Diagrams", Nordic Journal of Computing, Volume2, Number 4, p.p. 407-443, Winter 1995.

http://www.cs.helsinki.fi/njc/njc2.html#Parrow1995:407

[Penn93] John Penners, "Simple Mobile IP (SMIP)", Internet Draft draft-penners-mobileip-smip-00.txt, Internet Engineering Task Force (IETF), Sep 1993.

[PGLM97] Raj Pandya, Davide Grillo, Edgar Lycksell, Philippe Mieybégué, Hideo Okinaka, Masami Yabusaki, "IMT-2000 Standards: Network Aspects", IEEE Personal Communications, p.p. 20-29, Aug 1997.

http://www.comsoc.org/

[PJ00] C. Perkins, D.B. Johnson, "Route Optimization in Mobile IP", Internet Draft draft-ietf-mobileip-optim-09.txt, Internet Engineering Task Force (IETF), 15 Feb 2000.

[PJ98] C. Perkins, D. Johnson, "Mobility Support in IPv6", Internet Draft draft-ietf-mobileip-ipv6-05, Internet Engineering Task Force (IETF), 16 Mar 1998.

[PK98] Vu Anh Pham, Ahmed Karmouch, "Mobile Software Agents: An Overview", IEEE Communications Magazine, p.p. 26-37, Jul 1998.

http://www.comsoc.org/

[PRM97] Gian Pietro Picco, Gruia-Catalin Roman, Peter J. McCann, "Expressing Code Mobility in Mobile UNITY", Proceedings of the 6th European Software Engineering Conference (ESEC '97), Lecture Notes in Computer Science 1301, Springer-Verlag, Sep 1997.

[PT97] B. C. Pierce, D. N. Turner, "Pict: A programming language based on the pi-calculus", Technical Report CSCI 476, Indiana University, 1997.

[Q.1290] ITU-T recommendation Q.1290, "Glossary of Terms Used in the Definition of Intelligent Networks", Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T), Mar 1993.

http://www.itu.int/

[Rapel95] Juha Rapeli, "UMTS: Targets, System Concept, and Standardization in a Global Framework", IEEE Personal Communications, p.p. 20-28, Feb 1995.

http://www.comsoc.org/

[RF95] Pradeep Ray, Michael Fry, "Integrated Management in a Mobile Environment: A TINA Perspective", Proceedings of the Telecommunications Information Networking Architecture Conference, TINA '95, Melbourne, Australia, 1995.

[RFC0791] J. Postel, "RFC 0791: Internet Protocol", Internet Engineering Task Force (IETF), Aug 1993.

ftp://sunsite.doc.ic.ac.uk/rfc/rfc791.txt

[RFC1034] P. Mochapetris, "Domain Names – Concepts and Facilities", Internet Engineering Task Force (IETF), 01 Nov 1987.

ftp://sunsite.doc.ic.ac.uk/rfc/rfc1034.txt

[RFC1498] J. Saltzer, "RFC 1498: On the Naming and Binding of Network Destinations", Internet Engineering Task Force (IETF), Aug 1993.

ftp://sunsite.doc.ic.ac.uk/rfc/rfc1498.txt

[RFC1883] S. Deering & R. Hinden, "RFC 1883: Internet Protocol, Version 6 (IPv6) Specification", Internet Engineering Task Force (IETF), Dec 1995.

ftp://sunsite.doc.ic.ac.uk/rfc/rfc1883.txt

[RFC2002] C. Perkins, "RFC 2002: IP Mobility Support", Internet Engineering Task Force (IETF), Oct 1996.

ftp://sunsite.doc.ic.ac.uk/rfc/rfc2002.txt

[RFC2005] J. Solomon, "RFC 2005: Applicability Statement for IP Mobility Support", Internet Engineering Task Force (IETF), Oct 1996.

ftp://sunsite.doc.ic.ac.uk/rfc/rfc2005.txt

[RFC2041] B. Noble, G. Nguyen, M. Satyanarayanan, R. Katz, "RFC 2041: Mobile Network Tracing", Internet Engineering Task Force (IETF), Oct 1996.

ftp://sunsite.doc.ic.ac.uk/rfc/rfc2041.txt

[RFC2073] Y. Rekhter, P. Lothberg, R. Hinden, S. Deering, J. Postel, "RFC 2073: An IPv6 Provider-Based Unicast Address Format", Internet Engineering Task Force (IETF), Jan 1997.

ftp://sunsite.doc.ic.ac.uk/rfc/rfc2073.txt

[RFC2103] R. Ramanathan, "RFC 2103: Mobility Support for Nimrod: Challenges and Solution Approaches", Internet Engineering Task Force (IETF), Feb 1997.

ftp://sunsite.doc.ic.ac.uk/rfc/rfc2103.txt

[RLV95] Ramachandran Ramjee, Thomas F. La Porta, Malathi Veeraragahvan, "The Use of Network-Based Migrating User Agents for Personal Communication Services", IEEE Personal Communications, p.p. 62-68, Dec 1995.

http://www.comsoc.org/

[RM98] Gruia-Catalin Roman, Peter J. McCann, "An Introduction to Mobile UNITY", Proceedings of the 3rd International Workshop on Formal Methods for Parallel Programming: Theory and Applications (FMPPTA '98), in Parallel and Distributed Processing, Lecture Notes in Computer Science 1388, Springer-Verlag, Apr 1998.

[RMP97] Gruia-Catalin Roman, Peter J. McCann, Jerome Y. Plun, "Mobile UNITY: Reasoning and Specification in Mobile Computing", ACM Transactions on Software Engineering and Methodology, p.p. 250-282, Jul 1997.

[Rumb91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, "Object-Oriented Modelling and Design", Prentice Hall, Englewood Cliffs, NJ, USA, 1991.

[Sang93] Davide Sangiorgi, "Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms", PhD thesis, ECS-LFCS-93-266, Dept. of Computer Science, University of Edinburgh, 1993.

[Sang96] APM Ltd., "π-calculus, internal mobility, and agent-passing calculi", Theoretical Computer Science, Volume 167, Numbers 1 & 2, Elsevier Science, p.p. 235-274, Oct 1996.

[Sang98] Davide Sangiorgi, "Asynchronous process calculi: the first-order and higher-order paradigms (Tutorial)", To appear in Theoretical Computer Science, Elsevier Science, Dec 1998.

ftp://zenon.inria.fr/meije/theorie-par/davides/piHOPlasy.ps.gz

[Schul96] Henning Schulzrinne, "Personal Mobility for Multimedia Services in the Internet", European Workshop in Interactive Distributed Multimedia Systems and Services, Berlin, Germany, Mar 1996.

ftp://ftp.fokus.gmd.de/pub/step/papers/Schu9603:Personal.ps.gz

[Sewell99] Peter Sewell, "A Brief Introduction to Applied π", Notes from lectures at the MATHFIT Instructional Meeting on Recent Advances in Semantics and Types for Concurrency: Theory & Practice, Imperial College, Jul 1999.

http://www.cl.cam.ac.uk/users/pes20/mathfit-notes.ps

[Shoch78] John F. Shoch, "Inter-Network Naming, Addressing, and Routing", IEEE Proceedings of COMPCON, Fall 1978.

[TINA-overv] "Overall Concepts and Principles of TINA", Telecommunications Information Networking Architecture Consortium (TINA-C), 17 Feb 1995.

http://www.tinac.com/

[TINA-req] "Requirements upon TINA-C architecture", Telecommunications Information Networking Architecture Consortium (TINA-C), 17 Feb 1995.

http://www.tinac.com/

[TINA-SA5] "TINA Service Architecture, version 5.0", Telecommunications Information Networking Architecture Consortium (TINA-C), 16 Jun 1997.

http://www.tinac.com/

[TINA-SAA5] "TINA Service Architecture Annex, version 5.0", Telecommunications Information Networking Architecture Consortium (TINA-C), 16 Jun 1997.

http://www.tinac.com/

[Tirop98] Thanassis Tiropanis, "Offering Role Mobility in a TINA Environment", Proceedings of the 5th International Conference on Intelligence in Services and Networks, IS&N '98, Antwerp, Belgium, May 1998.

http://www.cs.ucl.ac.uk/staff/t.tiropanis/isn98rm.zip

[TMK99] Thanassis Tiropanis, Chris Malbon, Hervé Karp, "A Generic Component for Managing Service Roles", UCL Research Note, RN: RN/98/42, University College London, London, United Kingdom, Mar 1999.

[UMLv1n97] "Unified Modelling Language version 1.0: Notation Guide", Rational Software Corporation, 13 Jan 1997.

http://www.rational.com/uml/resources/documentation/notation/index.jtmpl

[UMLv1s97] "Unified Modelling Language version 1.0: UML Summary", Rational Software Corporation, 13 Jan 1997.

http://www.rational.com/

[UMTSmgmt] Draft ES 201 385, "Telecommunications Management Network (TMN); Universal Mobile Telecommunications System: Overview, processes and principles", European Telecommunications Standards Institute, Oct 1998.

http://www.etsi.fr/

[UMTSsat] ETSI Technical Report (ETR) 279, "Satellite Personal Communications Networks (S-PCN): Needs and objectives for standards in addition to the ETSs on essential requirements", European Telecommunications Standards Institute, May 1997.

http://www.etsi.fr/

[VITAL-D13] European ACTS project VITAL (contract number AC003), "ODTA Validation Third Trial, parts (A, B and C)", CEC Deliverable Number: AC003/SES/WP2/DS/P/D13/b1, The VITAL Consortium, Oct 1998.

http://www.mari.co.uk/vital/

[VM94] Björn Victor, Faron Moller, "The Mobility Workbench: A Tool for the π-Calculus", Technical Report ECS-LFCS-94-285, LFCS, Dept. of Computer Science, Edinburgh University, Feb 1994.

[VT97] J. Vitek, C. Tschudin, Eds., "Mobile Object Systems: Towards the Programmable Internet", Lecture Notes in Computer Science Series, Volume 1222, Springer, 1997.

[Yemini91] Y. Yemini, "Network Management by Delegation", Integrated Network Management II, Krishnan and Zimmer, Eds., Elsevier 1991.

Appendix A: CORBA design pattern template

In this section we give a concise description of the CORBA design pattern template by Thomas J. Mowbray and Raphael C. Malveau, based on their book "CORBA Design Patterns" [MM97]. We provide a short description for each field of the design pattern template. The fields that appear underlined in this description are mandatory.

Most Applicable Scale

This field describes the scale on which the design pattern is most applicable. The CORBA design pattern template defines the following scales in its scalability model, in bottom-up order:

S1. *Objects and Classes*: This scale is concerned with objects and object classes. It can describe solutions for the definition and management of object classes and object instances in terms of object attributes and signatures for operations between objects. Discussions on this scale are so specific that this scale is suggested for considering the specifics of a system's implementation.

S2. *Microarchitecture*: On this scale, design patterns combine multiple objects or object classes. At this level design patterns are used to tackle limited problems with software applications. The Gamma pattern language [GHJV97] is primarily concerned with design patterns of this scale.

S3. *Framework*: Here, problems at a macrocomponent level can be addressed. Design patterns of this level usually involve one or more microarchitectures. The goal is to reuse both software code and the design used in writing the code. At this level, the presence of certain capabilities within a system (such as an object request broker) is often presupposed.

S4. *Application*: At this level, design patterns tackle the structure, control and management of a single application program.

S5. *System*: This scale involves design patterns that add interoperation between applications. The system-level architecture is the enduring structure that survives the modification and replacement of component applications over the lifecycle of the system.

S6. *Enterprise*: This is the largest architectural scale within an organisation. Software at this level includes many systems, and each system includes many applications. This scale is concerned with establishing policies and procedures that apply throughout an organisation.

S7. *Global/Industry*: This is the highest scale, which comprises multiple enterprises. Design patterns on this scale address the impact of software that crosses multiple enterprise boundaries. Issues regarding languages, standards and policies that affect multiple enterprises are addressed here.

Solution Type

There are four possible types of solutions that a design pattern of this template can address:

ST1. *Software*: for design patterns that concern the creation of new software.

ST2. *Technology*: for design patterns that solve software problems through the adoption of new technologies and not by creating new software from scratch.

ST3. *Process*: for design patterns that provide a solution to a software problem as a decision-making process.

ST4. *Role*: for design patterns that solve software problems by allocating clear responsibilities to organisational stakeholders.

Solution Name

This is a unique identifier for the design pattern. The solution name is used for future reference to the pattern.

Intent

This is a brief statement of the problem that the design pattern attempts to solve. It should be a maximum of three lines.

Diagram

Diagrams can be used to clarify the solution that a design pattern offers.

Primal Forces

This field states the context of the choices that were made for a specific design pattern. There are one or more primal forces that outline the scope of a decision or a solution for any problem of any scale. Each of them sets the context according to which the specified problem is solved by the design pattern. Primal Forces include:

PF1. *Management of Functionality*: which expresses the primal force of making sure that a specified solution meets the end-user requirements.

PF2. *Management of Performance*: which concerns making sure that the software meets performance needs.

PF3. *Management of Complexity*: which involves analysing a design and making sure that the appropriate abstractions are made so that the design is simple and easily adaptable to future changes.

PF4. *Management of Change*: which involves identifying the areas where a system needs to be adaptable in the future and provide for future changes.

PF5. *Management of Information Technology (IT) Resources*: which concerns the impact of large scale on the ability to manage the assets of an enterprise. This force involves aspects such as hardware/software acquisition, inventory, training, maintenance, upgrade and support. One important aspect of management of IT resources is security management.

PF6. *Management of Technology Transfer*: which comprises some of the key forces that concern the transfer and dissemination of software and other technology from an enterprise. The control of intellectual property and the dependencies of internal systems on external technologies are some of the concerns here.

Applicability at This Scale

This field addresses the motivating factors for the use of the design pattern. It gives a number of factors, which, if they apply for a specific problem the design pattern is applicable. If a design pattern is applicable on more than one scales, factors for other scales can be given in the "Rescaling to Other Levels" field.

Solution Summary

This field describes how the problem, which is stated in the “Intent” field, is solved by the design pattern. This field should relate the forces that are identified in the “Primal Forces” field to the given solution.

Benefits

All the advantages of the solution are highlighted in this section.

Consequences

The undesirable consequences of the solution are stated in this section.

Variations to the Solution

This field provides extensions or variations to the solution of the design pattern that expand the capabilities of the pattern. How the benefits or the consequences of a design pattern are affected by applying these variations can also be stated.

Rescaling to Other Levels

Here, the efficiency of the design patterns on other scales is discussed. All the changes that apply to the capabilities of the design pattern when rescaling should be identified here.

Related Solutions

This field references other design patterns that resolve similar problems. It relates the current design pattern to the other patterns.

Example

This field describes an example of how the solution is applied to a particular problem.

Background

This field gives further examples of where the problem that the design pattern tackles occurs. It can also contain background information that is considered useful or interesting with regard to the design pattern.

Appendix B: Brief tour of the π-calculus notation

Names

We assume that there exists an infinite set of names which are usually denoted by lowercase letters: x, y, z, …

Action prefixes

Action prefixes (π) represent either sending or receiving a message (a name) or a silent transition in an agent.

$x(y)$ receive y along x

$\bar{x}\langle y \rangle$ send y along x

τ unobservable action (silent transition)

Agent or process expressions

π-calculus processes (P) are defined by the following syntax:

$$P ::= \sum_{i \in I} \pi_i . P_i \mid P_1 \mid P_2 \mid new \alpha P \mid !P$$

Where I is any finite indexing set and $\sum_{i \in I} \pi_i . P_i$ are summations or sums of action prefixes (π) followed by process (P). P^{π} is the set of π-calculus processes.

For every identifier of an agent $A(y_1, y_2, \ldots, y_n)$ with arity n, there is a unique defining equation $A(y_1, y_2, \ldots, y_n) \stackrel{def}{=} P$.

Polyadic π-calculus extension

This extension allows processes to exchange messages that contain more than one name. Receiving a message with multiple names y_1, y_2, …, y_n along x is denoted by $x(y_1, y_2, ..., y_n)$, while sending a message with multiple names y_1, y_2, …, y_n along x is denoted by $\bar{x}\langle y_1, y_2, ..., y_n \rangle$. The first formula can be encoded in the monadic π-calculus as $x(y_1).x(y_2)...x(y_n)$, while the second one can be encoded as $\bar{x}\langle y_1 \rangle.\bar{x}\langle y_2 \rangle...\bar{x}\langle y_n \rangle$.