INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Université d'Ottawa · University of Ottawa

Data Management for Supporting Nomadic Users Based on LDAP and Software Agents

by

Amin M. Hooda, B.Eng.

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Applied Science
in Electrical Engineering

Ottawa-Carleton Institute of Electrical and Computer Engineering
School of Information Technology and Engineering
Department of Electrical and Computer Engineering
Faculty of Engineering
University of Ottawa

October, 1998

Abstract

Today's road workers increasingly demand computing and communications capability that is available from any place, in any form, and at any time. Such network access primarily needs to be independent of location and device, which identifies 'global mobility support' across wired and wireless networks, popularly referred to as PCS in North America and as UMTS in Europe.

In the computing world, the emergence of the ubiquitous information access is noted as Nomadic Computing. The two types of mobility: terminal and personal, essentially characterize it. The thesis presents application layer solution, referred to as *Nomad's Personal Access System* (*NPAS*) that emulates 'personal mobility' in a virtual network, with the aim of supporting 'communication' needs of the mobile users.

The thesis proposes a data model that is required to provide personal mobility. The data model allows specifying the communication and computing environment, i.e., resource (services) profile of the user both at home and visited sites. As part of data management scheme, software agents is leveraged to create dynamic data pertaining to a current resource (services) profile of a nomad at a visited location. The data management involves the use of IETF's Lightweight Directory Access Protocol (LDAP) and inter-site negotiation agent called *Site Profile Agent*. The inter-site negotiation protocol is built using Knowledge Query and Manipulation Language (KQML). *NPAS* aims to deliver messaging services as the final promise; however, messaging services are only discussed to elaborate the use of the proposed data model.

Acknowledgements

I would like to extend special appreciation to my thesis supervisor: Prof. Ahmed Karmouch, for leading me into the innovative area of nomadic computing and for his continued guidance and support throughout this thesis work. I also want to specially thank Dr. Sue Abu-Hakima from NRC, to alert me more than eight months before the submission that I must not dive further, which rescued me from drowning. Especial thanks are due Dr. Mark Vigder of Sofware Engineering at NRC for listening to my day-to-day brainstorms. Thanks are also due members of SPIN group at NRC, for their many informal discussions and all my colleagues in Multimedia Research and Information Lab, at University of Ottawa, who have always been a source of motivation for their hard work and craving for learning. I particularly would like to thank, the current group leader and former comrade of SPIN, Dr. Roger Impey and SPIN's system admin, Randy Born for arranging to provide Dell's laptop at home for thesis writing, and then graciously extending the lending period for several times.

Life is not complete without family: I can never fully recognize the love and dedication of my mother (May God bless her soul). In absence of my parents (May God bless their souls), the constant support, affection, and prayers of my brothers and their wives, my nephews: Zarman and Amin, has been a great blessing in my life. I thank you all for being there for me through all times.

Last, but not least, I would like to acknowledge the Aga Khan Foundation, Geneva for giving me an opportunity of life through the AKF International Scholarship Award for the perusal of masters program.

Table of Contents

List of Figures

Chapter 1

1 Introduction

This research is concerned with the provision of seamless communication for the users that are roaming through a heterogeneous network environment. The following section provides a brief motivation for this work. The second and third section presents its precise objective and main contribution, respectively. This chapter is then concluded with the outline of the thesis.

1.1 Motivation

There is a pervasive, global desire to have network access anywhere, anytime and in any form. The desire has its roots in cellular technology, which in its current state has instilled in its users to expect communications support at any place and any time. Parallel to it, the progress in computing industry has created in the users' mind a need to do computing on-the-go by providing them with miniaturized, powerful portable and hand-held devices (e.g., PDA, palmtops, etc.) – but these still lag in communications capability.

Another strong driver for the pervasive communications access stems from multiple, yet disparate messaging forms that surround users today. Such as e-mail, voice mail, fax, pagers, and video-conference. These communication resources fall short on working in an integrated fashion. Since they have been designed and developed independently and with the specific service(s) in mind. However, users strongly wish that these discrete devices should be integrated to form a user-centric service environment.

Assume that a user is at a place, away from their home base, where they do not have the ability to receive phone calls. However, the user has access to e-mail and fax at the visited place. Despite access to e-mail and fax, user's communication ability is completely disrupted, since a calling party is dialing a user's phone at home base. It thus points to a characteristic of present-day telecommunications networks where people are bounded to communicate to devices, e.g., a person can only call XYZ fax machine at a company, assuming the destined receiver is there to collect the fax.

The ability to communicate while in transit, the availability of heterogeneous message modalities (e-mail, fax, voice, and video) and the need to have user-centered communication environment combined to spawn a novel vision of seamless communications. It is envisaged at the international fronts that seamless networks created by the integration of the telecommunication and data worlds, known in North America as PCS (Personal Communication Services) and in Europe as UMTS (Universal Mobile Telecommunication System), would provide personalized communications and some data services to mobile users irrespective of underlying networks. These services are attributed to the notion of personal mobility.

In the computing world, the eminence of such integrated global networks is noted as a radical shift in traditional computing towards, variously called mobile, nomadic or ubiquitous computing. The aim of nomadic computing is to offer personal mobility support to nomadic users i.e., anywhere, anytime, customizable, computing and communication access across heterogeneous networks. Due to the much broader perspective of nomadic computing, as is described later, in this thesis it forms the basis for deriving the full scope of functionality desired by the mobile users.

1.2 Objectives

The thesis work addresses the issues related to the data management for personal mobility. The main objective of the thesis is to design and implement the data management mechanisms to support the communications needs for those users who seek personal mobility in a private virtual network that spans multiple organizational networks.

1.3 Main Contribution

The main contribution of this research is the design and implementation of a data model and the agent-based data management mechanism for supporting communications needs of roaming users in a mobile computing environment. In this research other system-level components, required to support seamless communications, are also identified.

The data model provides the attributes to define the communications environment of the user both at home and visited site. It is strongly recognized here that the data model acts as the basis for the creation of messaging applications that cater to the communication needs of the nomadic users. Furthermore, to produce the dynamic data due to the changing location of a nomad, the software agent concept is leveraged. Namely, *Site Profile Agents* posted at home and visited sites perform the inter-site negotiation between the two sites to assign a new guest profile also called virtual or surrogate home for a nomadic user. This negotiation is conducted in view of several criteria such as policies of the visited site, requirements of the nomadic user, etc. The

negotiation protocol is defined using Knowledge Query and Manipulation Language (KQML), which is designed by ARPA Knowledge Sharing Effort [19].

This research is a part of the larger project of *Mobile Agents Alliance* (*MAA*), a collaborative effort, that includes the University of Ottawa, the National Research Council of Canada, and Mitel Corporation. The validity of this architectural design will be demonstrated in the three-site personal mobility testbed of *MAA*. The prototype implemented for data management mechanism is to be integrated with the larger project of *MAA*.

1.4 Thesis Outline

The rest of the thesis is structured in the following manner. Chapter 2 provides an insight into various research and development activities related to the area of personal mobility. Chapter 3 identifies and discusses the issues that are addressed in this thesis and describes the incremental approach adopted, catering to the ultimate goal of seamless communications. Chapter 4 presents the concepts, analysis and the design bits of the solution developed in this thesis, which is referred to as *Nomad's Personal Access System* (*NPAS*). The chapter provides elaborate description of *NPAS's* architectural components. Chapter 5 discusses a detailed design and the implementation of data management components of *NPAS*. Chapter 6 presents the summary and conclusions of this thesis and provides suggestions for the future work.

Chapter 2

2 Overview of Personal Mobility

This chapter presents an overview on the rapid progress made in the area of mobility. Firstly, a brief introduction is presented to highlight the diverse network resources, afforded by the existing environment, emphasizing the heterogeneity and interoperability problems of the existing networks. It stems the discussion of two fundamental characteristics of converged networks of tomorrow i.e., terminal and personal mobility support. Then, the various issues related to terminal and personal mobility, that have been identified and addressed by the research and development communities, are highlighted. Some examples of the future and current mobile applications are given at the end.

2.1 Introduction

The progress in portable computing and wireless communications has given in the hands of users diverse mobile devices, e.g., personal digital assistants (PDAs), laptops, cellular phones, pagers, and so on. Moreover, today computing devices are also enabled to access networks through wireless interface. However, wireless devices are still constrained to roam within the region of a single network, primarily due to different wireless environments and incompatible standards. To expand the roaming range through heterogeneous wireless networks, a terminal requires multiple network interfaces. As an example, consider the case of a user who goes from an indoor wireless LAN (IEEE 802.11) to outdoor wide-area data network (Advanced Radio Data Information Service - ARDIS, Celluar Digital Packet Data – CDPD). Due to the disparities in modulation

schemes, frequency bands, data rates, etc. for indoor and outdoor wireless transmission, the network interface cards needs to be switched. Furthermore, heterogeneity is encountered across data and telephony wireless networks (e.g., WLAN, wider-area data networks, cordless telephony, cellular mobile, and satellite mobile). For example, heterogeneity prevails when a user on the Internet aspires to connect to a user on Public Switched Telephone Network (PSTN). Thus, it calls for interworking at different layers including physical, data link, and network layer.

Interoperability between packet-oriented data networks and circuit-switched telephony networks of today and tomorrow is of central importance for establishing a global network [1]. A global network would allow seamless connectivity across wireline and wireless architectures of these two types of networks. Establishment of a global, integrated telecommunications and data network is being actively deliberated at the international fronts. This has been known in North America as Personal Communication Services (PCS) and now formally as International Mobile Telecommunications 2000 (IMT-2000), and in Europe as Universal Mobile Telecommunications System (UMTS). Figure 2.1 illustrates a high-level impression of currently evolving integrated heterogeneous network.

This emerging convergence activity has spawned a paradigm shift in the computing world towards the vision of integrated computing and communications networking support, called nomadic computing [2]. The nomadic computing aims to provide a rich set of computing and communications services to mobile users across heterogeneous interconnected wired and wireless networks in a transparent and convenient way. Regardless of wireless access, the users in a nomadic environment are

able to stay connected while they move from one location to another, and typically the access is through changing devices. Here, the users are in control of their accessibility. Therefore, network cannot relay those messages (phones, e-mails) that are less important, when a user is on a low-bandwidth, expensive connection in a hotel abroad, for example.

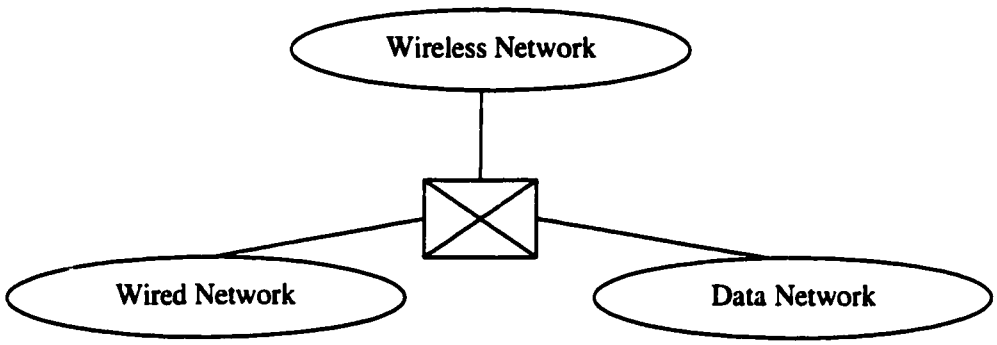


Figure 2.1: Architecture for Integrated Heterogeneous Networks for PCS services

2.2 Definition of Terminal and Personal Mobility

It is believed here that the two types of mobility, i.e., traditional terminal mobility and the more challenging personal mobility characterize Nomadic computing. The definitions of these characteristics substantially differ in telecommunications and computing world. Nomadic computing, due to its broader perspective of existing heterogeneous networking infrastructure, has provided a larger set of functionality.

Advances in radio and wireless access technologies enable the traditional terminal mobility. Which refers to the ability of a hand-held device that can access mobile communication network within a limited range, where a device relies on the network to locate and identify it as it moves. The nomadic environment not only extends radio-range of a mobile terminal to a global scale, but also supports a computing terminal without a wireless interface that hooks-up onto wireline network from different locations without

changing its terminal identity. Further, radio terminals are also enabled to afford multimedia communications. Thus, many of the network-based desktop applications (e-mail, web browsing, electronic catalog shopping, etc.) will be provided at the user terminal.

With terminal mobility, calls are still delivered to devices not to subscribers. In addition, solely terminal mobility support is not useful if the subscriber is exposed to more than one networked device. Since, devices are not integrated to form a user-centered service environment. Therefore, a caller dials different numbers to reach these devices. Moreover, the features assigned to one device, for instance, savings plans, message-filtering agents, etc. are not applicable to another device.

Since, the evolution of mobility is towards personal mobility; it is viewed as the next generation service and is being now actively sought both in the computing and telecommunications world. It is aimed to provide personalized computing and communications support regardless of device, location, and underlying network. A subscriber of personal mobility can engage in computing or/and communications (as an originator or a recipient) on any fixed or mobile terminal in any location. Independence from user's personal portable device, like PCS phone or laptop, cannot be supported in terminal mobility without disrupting the information access capabilities of the owner. As compared to terminal mobility, in personal mobility users have increased ability to choose and modify services environment. For example, not only users can define their accessibility to others, but also dynamically change their preferences of devices. The ability to maintain a user's service profile or work environment (device preferences,

privileges) despite changing locations is also known as service mobility, whereas in this thesis it is recognized as an intrinsic characteristic of personal mobility.

Terminal mobility is considered as a special case of personal mobility, where a nomadic user is carrying a portable mobile device and the underlying network is a wireless or wireline one. The figure 2.2 below visualizes the difference between terminal and personal mobility.

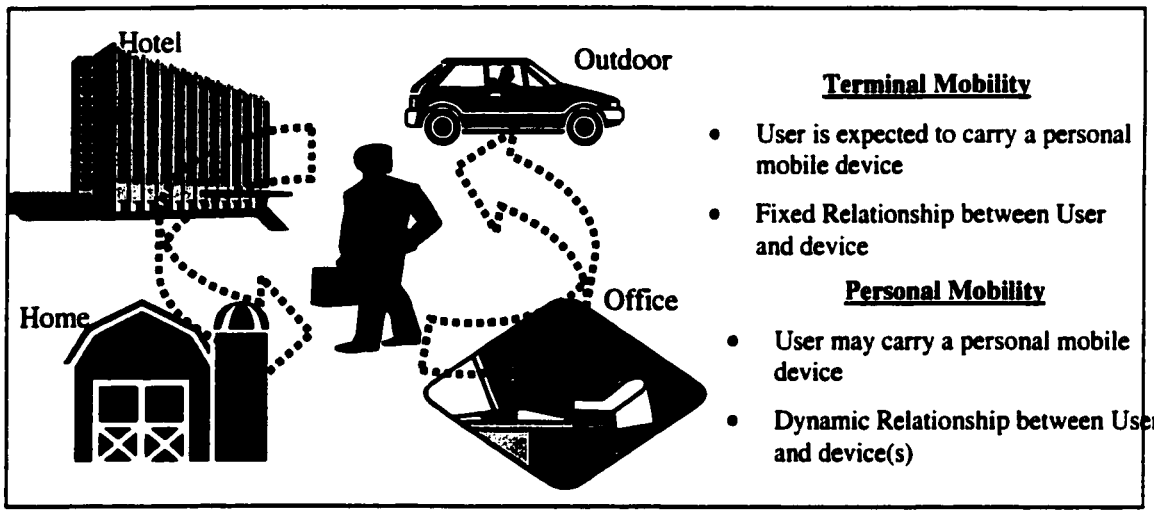


Figure 2.2: Terminal vs. Personal Mobility

2.3 Discussion of Issues Related to Terminal Mobility

The wireless interface lends mobility to a portable computing or communications device; however, it does not ensure transparent, anywhere and anytime network access. It inherently poses connectivity failures and bandwidth constraints due to interference and the limited frequency spectrum. Example: a travelling business user who tries to access to the Internet through wireless modem faces several constraints: the bandwidth provided by wide-area wireless medium is significantly less than that of the LAN at a user's home

base; the connection time is more expensive; certain files or applications required en route are not stored a priori. These issues are addressed by mobile file systems (CODA, ODYSSEY [4]), using predictive file pre-fetching and by exploiting weak-connectivity. Similar limitations would take effect if a traveler were to hook up their laptop through a dial-up connection at a hotel. Therefore, it is noted that wireless mobile computing does not entirely represent the issues of terminal mobility, as recognized in the nomadic computing research community.

In addition, since a computing device can be mobile, the current IP scheme is not suitable. In the current IP scheme, the IP address is inextricably tied to a network location. Therefore, when a mobile host changes its location from one domain to another, its IP address must change. Subsequently, IP packets addressed to the old address of a mobile meet a spurious disconnection. To address this issue, in 'mobile'-IP [5][6], each mobile host is assigned a Home Agent. As the mobile host moves, it registers to Home Agent via a Foreign Agent. When the Home Agent receives packets destined to the mobile host, it tunnels them to the serving Foreign Agent, which forwards them to the mobile host.

There are other issues including security, user interface and finite energy constraints associated with the size and weight of mobile terminals, that are crucially important to the widespread acceptance of terminal mobility.

2.4 Discussion of Issues Related to Personal Mobility

Unlike terminal mobility, personal mobility affords a greater degree of freedom by liberating the nomadic users from device dependency and by integrating user's

multiple devices. To attain the merits, many challenges need to be addressed, some of which are highlighted here; and more details appear in the chapters 3 and 4.

In the existing networks, communication is location-centric; that is, individuals can communicate only in terms of locations of devices. For example, a caller is bound to call a phone number in person's office or kitchen, whereas personal mobility offers to communicate directly to the person, not to their XYZ machine at the company or home [7]. This requires that each user be assigned a unique personal identifier or name, so that a calling party can connect to the person. This ability provides independence of location and devices, rendering unrestricted network access at any location.

Assume that a nomadic user visits a location that supports personal mobility. By the definition of personal mobility, a visiting user is able to use the computing and communications resources, e.g., telephone, fax machine, and network computer, in that location in conformance with its rules. This leads to the issue of on-the-fly discovering resources in a visited location and intelligently associating the user with the resources.

As indicated earlier, in personal mobility multiple devices are combined to form user-centered, integrated service environment. It means that if a user is called using their unique personal identifier via a telephone, the incoming message may arrive via another modality, e.g., e-mail, for the user who does not wish to receive any phone calls at a particular time. Associated to it, is the ability to customize services attached to devices, based on a user's computing (data and text files, view of the file directories, bookmark file, favorite editor, browser, etc.) and communications (phone, fax, e-mail, etc.) profile.

In the context of computing world, personalization or customization adds a new dimension, i.e., when a user moves from one computing terminal to another, they would seek the same interface, the similar environment as is on their previous fixed or mobile terminal. This requires porting the user environment from one computing terminal to another. All the issues identified above must identify and authenticate a nomadic user; therefore, security is an increasingly important, cross-issue that comes before and during any service provision session. In the following section, the currently known efforts pertaining to a few of the personal mobility issues are presented.

2.5 Efforts Addressing Issues of Personal Mobility

Since, more efforts have been concentrated in terminal mobility – presuming it to be as a complete characterization of mobility - much work supporting personal mobility has not been accomplished. Personal mobility has more recently been considered as a natural extension to terminal mobility. Nevertheless, personal mobility has been taken as a crucial service of global integrated network where it is noted as Universal Personal Telecommunications (UPT). In this context, the solutions reported here represent primarily the on-going research work in the area of personal mobility.

The issues specific to communications support for nomadic users seeking personal mobility, as stated at the outset, is the focus of this thesis. Thus, they are discussed in the chapter 3 of the thesis. This section aims to present the solutions solely pertaining to the provisioning of computing support to users as they move from one terminal to another.

2.5.1 User Environment Migration

In [9], the author has identified a novel challenge of process migration to follow a user's moves to provide ubiquitous computing environment. The paper considers three types of mobile entities: users, views and platforms. Users are defined as mobile. A view is what users see at the display screen of their PDA or workstation. The view is defined by user's environment setting and it includes the knowledge of local and remote external resources attached to the applications, which may be executed by the user. A platform implements a view as a collection of software and hardware. The platform software includes active threads that implement the view, the code that implements software environment in which code is executing (e.g., Java Virtual Machine), etc.

The aim of the author is to support: (i) as users move from location to location, their view should be portable, permitting a user to continue whatever work was being performed at the previous platform; (ii) as users move with a platform, in which case, the view stays consistent. However, if the application running on the platform has a network connection, it may be severed and therefore, be reconnected at a new site, which is similar to process migration. For the view mobility, i.e., when a view migrates from one platform to another, the author proposes two solutions and then compares them. Either the threads and data related to the view must migrate, or the threads must be notified of the location of new view and the I/O must be redirected to and from the new platform. The paper also points out the problem of user identification and authentication as one that is required before providing view or platform mobility. Heterogeneity of hardware architecture is considered as an acute problem, and an interpreted programming language is proposed. Then, the author discusses few of the mobile agents' middleware, e.g.,

Aglets, Telescript. As these mobile middleware platforms cater to the need of the view mobility, by exploiting transportability of agents.

2.5.2 Virtual Network Computing

The problem addressed in [10] is to allow a user to access their home computing environment on a personal Unix or PC desktop from any location. The proposed solution is called Virtual Network Computing (VNC) protocol, which is implemented as a client-server protocol for remote access from a simple, inexpensive client device. This protocol not only imitates the applications and data but also the entire desktop environment from the server machine. According to the authors, "whenever and wherever a VNC desktop is accessed, all its configuration and state (right down to the position of the cursor) are exactly the same as when it was last accessed". It also allows a single desktop to be accessed concurrently by several users, thus permits application sharing. The VNC is completely independent of operating system. Moreover, it is promoted as a thin-client system. And, therefore, it claims to run on a wide range of hardware.

2.6 Mobile Applications

Nomadic computing provides a new category of applications that are aware of the context in which they run [11]. Context includes the user's location, lightning, noise level, network connectivity, communication costs, bandwidth, the social situation; e.g., who the user is currently with, a manager or a colleague. Based on the dynamic context, terminal adapts to the environment. For example, some important applications are those that provide crucial information to workers on the road, or in remote or dangerous places. For example, the information pertaining to finding the way, exploring nearby people or

resources, and avoiding pitfalls. These applications also represent information services, as local yellow pages, possibly extended with on-line information such as movies currently playing at local theaters. The context-aware applications also provide electronic news services tailored according to the individual user profiles. It should be noted that context-aware applications take into account a user's accessibility to computing and communication devices. Therefore, these applications will exploit user's integrated service environment.

Another class of applications is mail-enabled applications [12]. Users carrying personal communicators will be able to initiate and receive messages from any location. They will be able to receive alerts about pre-defined conditions (such as plane delay or heavy traffic on the way), which is irrespective of time and location.

A number of vertical applications for terminal mobility already exist. These include the FRIEND system, or First Responder Interactive Emergency Navigational Database, which was developed in conjunction with the Bellevue Police Department in the North Hills of Pittsburgh [13]. It provides distributed wireless access to different levels of emergency staff. Emergency monitoring personnel could be the first responders (police, fire, and emergency medical). The FRIEND has the ability to handle a flood of information generated by multiple users during the emergency. The FRIEND facilitates collaborative work, as multiple users can know each other's positions and the common strategy can be easily followed. Another example of terminal mobility application is MediPad project at Purdue Uinversity [14]. MediPad system provides up-to-date information on patients, while they are not at the hospital, through a mobile device. The specialist medical staff carrying MediPad mobile device can be reached promptly to

rescue lives. Some other vertical applications are pen centric data entry devices (such as carried by United Parcel Service personnel), mail tracking, real-time support for travelling sales personal. Yet, another existing application of mobile computing is the so-called Active Badge technology, where infrared communication is used for locating employees and redirecting voice mail and data.

Chapter 3

3 Intentions and Approach of this Research

Having introduced the context of personal mobility in previous chapter, a precise elaboration of the problem: data management for personal mobility users, addressed in this research, is provided on below. Following that, the approach adopted here to address it is discussed. Finally, a comparison is given, where the distinctive elements of this research are highlighted.

3.1 The Problem

This research aims to design and develop a software solution to provide an integral function of personal mobility, i.e., supporting the communications needs of the users who move between different locations. The assumption here is that locations are associated to different organizations. To comprehend the scope of this problem, the following is noted:

- Generally, the users may have access to more than one communication medium, for example, e-mail, phone, fax, etc. Whereas, at times they may only have access to a single communication device depending on a location.

- The locations at different organizations have varying types of services to offer and they are managed through a set of policies. The resources usage (load conditions) at these locations would affect the availability of those services.

- Users do not want to be deluged by junk communications. Therefore, the users should be allowed to easily modify their reachability. The reachability here refers to the incoming messages that could be directed to the user. Which incoming messages should be forwarded and what modality is preferred in a particular situation (meeting, office, etc.) are some of the elements that define reachability.

- Currently, the different communication modalities (e-mail, fax, voice) do not interoperate due to the various reasons, including (i) each modality was developed independently; (ii) today's network functions like addressing, routing and management are centered around devices and not users; and (ii) there is a lack of integration between heterogeneous networks (the Internet, PSTN, wireless networks).

- The available network infrastructure and the corresponding devices should be leveraged to provide anywhere, anytime communication support. Therefore, communications with any other person should be accomplished by using the existing modes of addressing i.e., telephone number, fax number, e-mail address, etc. Example: the calling party should be able to communicate to the called party, irrespective of the called party's location or device. Such communication is of course, screened in the light of the called party's profile, policies of visited site, etc.

The above issues indicate the nature and intensity of the challenge faced in achieving seamless communications for roaming users. The ultimate ambition is to provide an intelligent, customizable and integrated messaging environment for a user, which blends the available heterogeneous modalities, and dynamically builds the association of devices to users in correspondence to a user's changing locations. It is

deliberated here that this ambition translates into the two essential issues: (i) data management mechanisms and (ii) messaging applications for roaming users. The data management refers to the procedures that are required for identifying user, location, devices, types of services, policies, security restrictions, etc. Messaging applications address the communications needs of mobile users, for example, reachability. These applications rely on the data obtained by the data management mechanisms to deliver the communication support.

This work aims to support the communication needs of roaming users. Therefore, the messaging applications are of great interest here. However, the goal is to design and develop the components and mechanisms of data management that are required for supporting messaging applications. Thus, the approach here is to take a high-level view of messaging applications to determine the requirements on the mobility-related data and its mechanisms.

3.2 The Approach in this Research

Today's telecommunication and data worlds (until the realization of PCS) are considered disconnected and self-contained. Therefore, the design approach to address the problem stated above is carefully chosen to be incremental. Consequently, the replacement of or addition to the legacy user equipment is not required to provide seamless communications. This results in extension of the useful life of investments in the existing resources, and allows friendly penetration of novel services in the existing user environment. Further, it is not assumed that a nomadic user can be reached anywhere within its coverage by means of a universal personal number as opposed to UPT [17].

This means that as usual a caller, for example, dials a telephone number of the called party, and the call is delivered the same way as it is normally. However, this incoming call is intercepted at the home side of the called party where the capability to intelligently forward the call to the called party's current location is provided, as illustrated in figure 3.1. Intelligent forwarding refers to a decision taken in respect of a user's profile: user's preferences, messaging requirements and visited site permissions, restrictions, etc. In case of an outbound call (initiated by a visiting user), device-based user authentication is required. If such a means of authentication is available, a visiting user's call, in a general sense their actions, can be permitted or restrained in the light of the visited site's policies and the service profile of a visiting user.

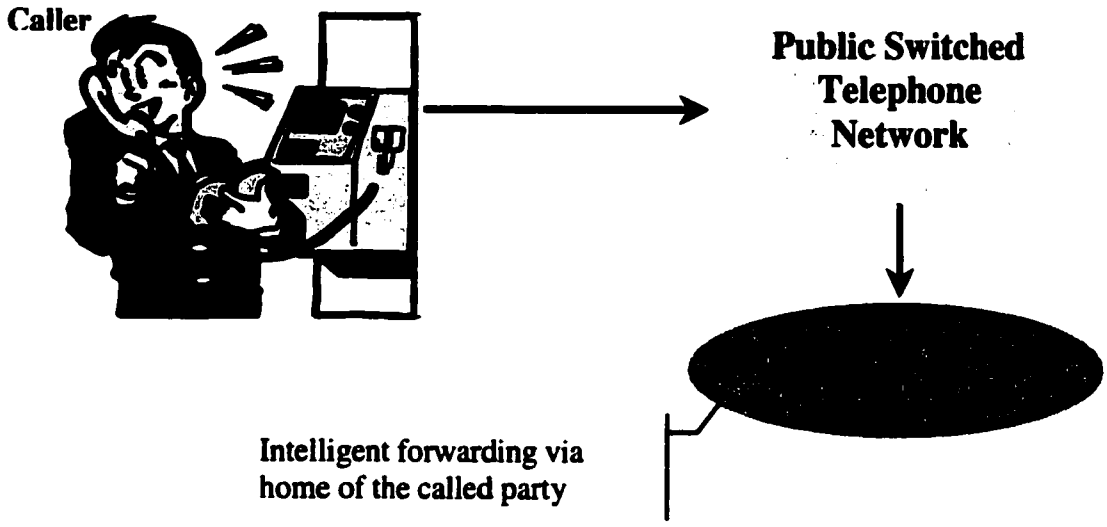


Figure 3.1: Intelligent Forwarding via Home of the Called Party

It should also be noted that the approach adopted here does not rely on electronic location tracking such as active badge technology, smart cards, or facial recognition (used in access control systems). Therefore, the onus of updating the user's current location is

on a nomad. For which a user must have an individual identification to inform the network environment of their whereabouts; therefore, their e-mail address is selected for this purpose. An e-mail address is assumed to uniquely identify a nomad at home and visited locations. The use of e-mail address here identifies with the incremental approach adopted. It allows the use of the existing id, and therefore, it does not impose on users to comply with a new requirement.

3.2.1 Open Data Network

The thesis work draws its inspiration from nomadic computing [2]. Since, nomadic computing has been conceived with the most comprehensive mobility support by including both the computing and communications services for roaming users. Namely, it includes the services and objectives of the Internet community, the cable, the telephone, and the entertainment industries. It emphasizes an open access by service and network providers. The vision of nomadic computing goes well beyond the capabilities envisaged by PCS and UMTS. The architecture for nomadic computing is envisioned as Open Data Network, which consists of four layers as illustrated in fig. 1; adopted from [18]. The four layers: the bearer service, transport, middleware, and the applications are briefly described based on [18].

- The bearer service sits on top of the network technology substrate: the range of technologies that realize the raw bit-carrying fabric of the infrastructures. It is an abstract bit-level transport service. Therefore, this layer allows interchange of information and services across different underlying networks. It also implements a specified range of qualities of service (QoS) to support the higher-level services

envisaged for the ODN. At this level, bits are bits. Nonetheless, having multiple QoS permits an application with a particular service requirement to make a suitable selection.

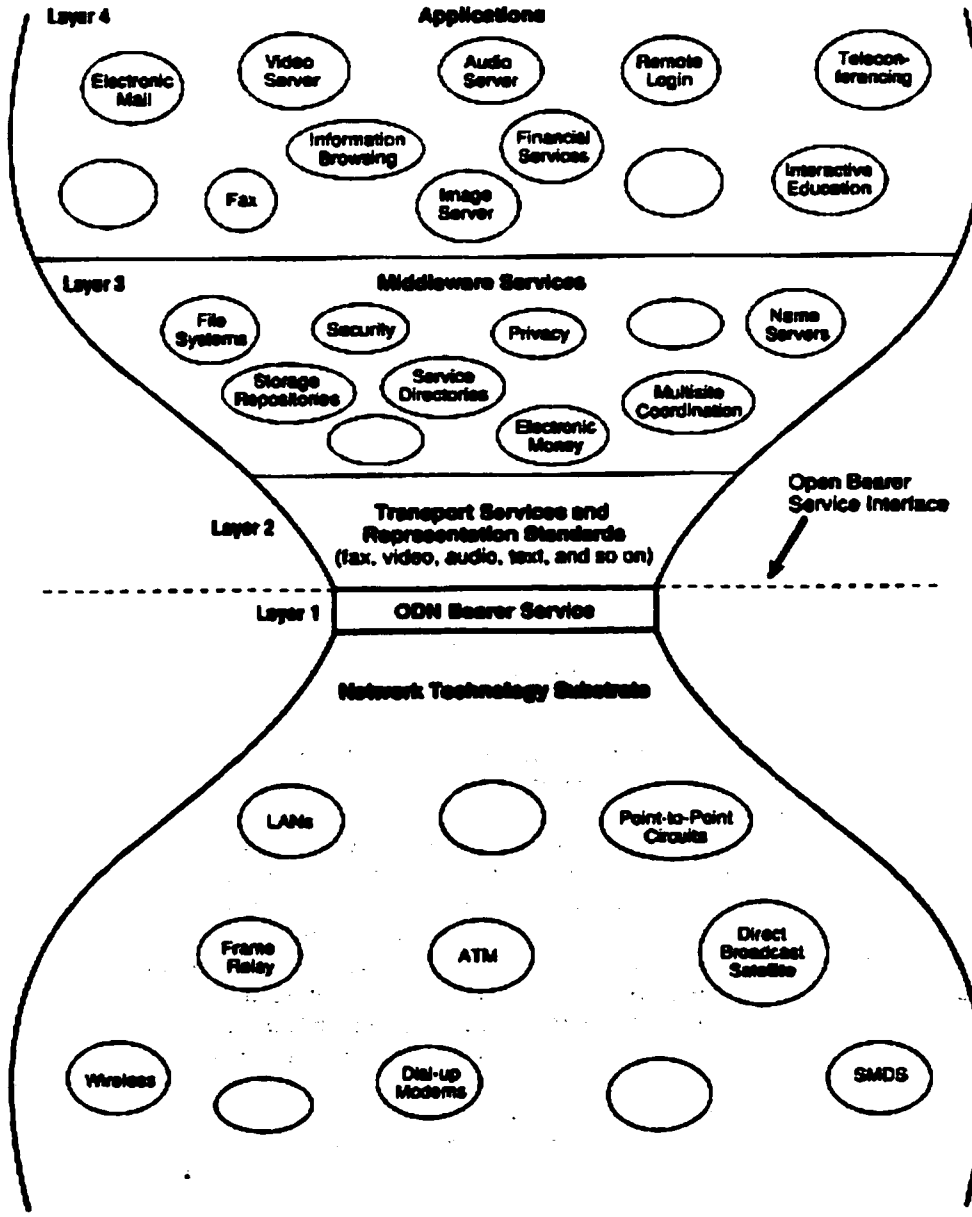


Figure 3.2: A Four-Layer Model for the Open Data Network

- At the next level is the transport layer. It manages end-to-end delivery services needed by the applications. Service features at the transport layer include reliable, sequenced delivery, flow control, and end-point establishment. The bit-streams of bearer service can be differentiated at this layer into identifiable traffic types such as voice, video, text, fax, graphics, and images. Many of the services offered at this level are well understood and form a mature aspect of networking today.

- Much of the action for nomadic computing takes place at the middleware level of the commonly accepted layered architecture [2]. It is composed of a set of higher-level services, which provides an environment more directly suited to the advanced applications that run there. Examples of these functions include file system support, privacy protections, authentication and other security functions, tools for coordinating multi-site applications, navigation and filtering tools, remote computer access services, storage repositories, name servers, network directory services, and directory of other types. A subset of these functions, such as naming, will best be implemented in a single, uniform manner across all parts of ODN. Also needed at this level are tools for buying and selling products (goods and services), including a definition of electronic money, and an architecture for dealing with intellectual property rights.

- The uppermost layer contains the applications recognized by typical users. For example, electronic mail, airline reservation systems, systems for processing credit card authorizations, or interactive education. The benefit of the common services and interfaces of middleware layer is that applications can be constructed in a more modular manner, which should permit additional applications to be composed from these modules.

3.2.2 Open Data Network and the Approach in this Research

Although, the scope of this research is limited to the communications needs of a roaming user, the enhanced definition of personal mobility used in this research is derived from characteristics of nomadic computing. Further, the notion of dynamic association of users to devices, adopted in this research, is also broadly introduced in nomadic computing. A concrete design and implementation of it is given in this thesis. It is also found in this work that the services and capability set composed in the middleware layer of ODN call for the applicability of software agents, as described on below. Finally, the liberties to look into new concepts, new services and new technologies have been only possible due to the approach adopted here i.e., to benefit from a broad perspective of nomadic computing, which broadly introduces various aspects of mobility.

This research does not investigate any of the issues of the lower two layers of the ODN: the bearer and transport layers. The underlying physical networks in this thesis are assumed to be the existing ones; and therefore, the approach here is an incremental one, as defined above.

3.2.3 Software Agents for Nomadic Environment

In view of the architecture of nomadic computing, this section discusses the applicability of agents to achieve the services and capability set composed in the middleware and applications layer of ODN. Currently, "agents" is evolving as a powerful software paradigm. It is popularized as a software design and implementation tool, addressing various user and network-related issues that are complex, distributed, and dynamic in nature. The issues include: developing flexible and smart user-interfaces, providing efficient information retrieval mechanisms and creating electronic marketplace

over the Internet, performing advanced service provisioning in next generation telecommunication networks (Personal Communication Services - PCS, Universal Mobile Telecommunications System - UMTS), and supporting plug-and-play networks. Generally, software agent is understood as a program that acts on behalf of other entities in an autonomous fashion, which may carry some degree of proactivity, reactiveness, some level of learning, planning, negotiation, mobility, etc. Proactivity identifies a quality that generates an action or behavior before the occurrence of actual stimulus to that action or behavior. In contrast, reactivity generates an action or behavior only when there is a stimulus.

Regardless of it that the software agents be programmed as static or mobile code, they require a means to communicate amongst each other. The means of communication can be either procedural or declarative [19]. Based on the [20], it is easy to understand procedural approach as a direct interaction and declarative approach as an indirect interaction. In procedural or direct interaction, agents call directly the code within other agents. Scripting languages, such as TCL, Apple Events, and Telescript, are based on it [19]. Alternatively, when agents send messages to each other without knowing their object-dependent interfaces, it is referred to as declarative approach or indirect interaction. Here, the messages are represented in a common language, which is rich enough to express a wide range of information, such as, definitions, goals, and assumptions, etc. Its example is the notion of agent communication language (ACL) that is designed by ARPA Knowledge Sharing Effort [19]. Further, the difference in direct and indirect interaction is programming/scripting languages (e.g., Tcl/Tk, Java, Telescript) do not offer semantics of ACL that can express needs, beliefs, expectations,

intentions, capabilities, etc. [19]. This semantics is obtained through the ability to interpret the content of a message.

An ACL is a layered abstraction of the communication problem between heterogeneous software applications [21]. The heterogeneity in applications may occur due to several reasons, including, coding in different programming languages; using different conventions to refer to the same concepts, entities or objects; creating application-dependent primitives to communicate, or adopting a different content representation. A communication language is not concerned with the physical exchange of a message, as opposed to TCP/IP, CORBA, COM, etc. Nevertheless, the distributed object bus like CORBA, ILU, OpenDoc, apart from providing a transport mechanism allow applications to exchange data structure (usually objects) and methods across disparate languages/platforms - overlapping with one of the aims of ACL. However, ACL is distinguished from inter-language bridges because it promises the following:

(i) The meaning of the same concept or objects is understood uniformly across applications using a common conceptualization of a domain;

(ii) A large variety of message types/communication primitives (e.g., GET, POST in HTTP; get_request, set_request in SNMP) is created catering to the needs of various actions performed by software applications, providing a substrate layer for specifying negotiation or messaging protocol for applications;

(iii) Different content representations can be translated into a universal content representation language.

Therefore, adopting an ACL as a common communication language addresses the problem of interoperability and sharing of knowledge for heterogeneous software. The Knowledge Sharing Effort (KSE) resulted in an ACL that contains three parts: a definition of the domain for mutual understading (e.g., Ontolingua), an inner language called KIF (Knowledge Interchange Format - Interlingua) for content representation, and an outer language called KQML (Knowledge Query and Manipulation Language), which provides an intention of the content. Similar to KIF, Electronic Data Interchange is another example of content representation language. Although, it is not an AI-style language; however, it serves the same purpose [22]. The current wave of XML, the Extensible Markup Language presents another choice for content interchange format for networked systems.

To understand the level of interoperability offered by the KQML-speaking agents, consider a situation at a company that requires a system that can schedule group meetings, according to the availability of employees and locations [23]. Assume that the company already has: (i) employees who keep calendars in their personal computers; (ii) a database that stores information on the employees, such as names, offices, phone numbers; and (iii) a database that registers conference rooms, with additional information regarding capacity, availability, scheduled activities and so on. A common approach, i.e., bottom-up, is to build an application from scratch, so that one application holds all the necessary information and knowledge. The alternative is to use existing applications, i.e., top-down approach. Doing that would require: 1) the applications to comprehend each other's knowledge stores, despite differences in implementation languages and knowledge representation schemes, and 2) the applications to communicate with each

other and dynamically make queries, answer them, assert or remove facts from their knowledge stores, in short, to interact intelligently.

The top-down approach only highlights an instance of the issue that is addressed by KQML-speaking agent. These agents provide a software infrastructure for a larger problem of interaction and interoperability in the existing heterogeneous applications, which may be programmed in different languages, or developed using different databases or knowledge-base systems – an innate problem of the software world. Thus, agent-oriented programming makes a good tool to implement the middleware services of ODN, where different services may require cooperation and interaction with each other. These services include privacy protection, authentication and other security functions, coordinating multi-site applications, navigation and filtering designs, which could use agents on their behalf for achieving interaction among services. Nonetheless, according to [18], the development of middleware services is not as mature as the work on the other layers of ODN. Since, it can be noted that abstracting horizontal characteristics or services, that support a wide range of higher level services/applications visible to the user, is a complex issue.

3.2.4 Location Management Technique and the Approach in this Research

The existing wireless technology, being one of the enabling predecessors of PCS, also provides context to the work accomplished in this thesis. The wireless telephony world has received much attention from the distributed computing and telecommunications research community in the area of *location management* [24] for wide-area wireless access and recently for global access. Traditionally, *location management* is referred to as location updates, searches, and database look-ups for a

service profile of an active, roaming wireless terminal. Such *location management* attempts to support hundreds of millions of users in real-time (for synchronous communications) over bandwidth-limited wireless links. Thus, it has produced *Location Management Techniques* (LMTs) that are highly scalable and efficient [26].

The goal of traditional LMT*s* to locate active roaming device contrasts with the LMT for personal mobility, where the aim is to locate people. Nevertheless, the basic requirement to find the location of an object, be it a device or a person, is an overlapping aspect in both the terminal and personal mobility. Therefore, it is assumed here, for the first time, that traditional LMT*s* will strongly influence the evolution of new LMT*s* for personal mobility. Differences will be identified during the design of the *NPAS* in the next chapter.

3.3 A Comparative Look at Other Approaches

3.3.1 Universal Personal Telecommunications: An Intelligent Network Service

At the global level, the International Telecommunication Union-T (ITU-T) provides recommendations for telecommunication requirements related to the wired world. According to the ITU-T, Universal Personal Telecommunication (UPT) is an ability to reach a user anywhere in the world and to provide personalized and convenient communication access to the user. It introduces the novel concept of personal numbering that allows a user to identify them to network by registering at any networked device. Thereby, it provides the ability to route call to any of the multiple devices (fax, e-mail, phone, etc.) based on the user profile that contains pre-defined, explicit filtering and forwarding instructions. Finally, it gives the capability to modify dynamic features in the

user profile like call screening at the time of call delivery and to change static features in the user profile like saving plans, etc. [17]

UPT has given a boost to the Intelligent Network, the ITU-T standard, by identifying IN as a platform on which UPT would be built as an advanced service application. The reason IN is selected is that it provides a set of flexible, service-independent building blocks that separates the service control from the call switching. IN distributes computational load - intelligence/services - into functional entities (Service Switching Point, Signaling Transfer Point, and Service Control Point). However, IN does not allow information transfer across SCPs. Since interaction between SCPs is not defined, it restricts service mobility across different network operators. Therefore, IN requires introducing mobility functions in the fixed network [27], [28], [29], [30]. IN is also attractive to the wireless world due to its faster service creation capability and deployment of new services with a minimal impact on existing networks. In its current form, it is not fully integrated with mobile networks, which incurs a longer processing time for the provisioning of an IN service [29],[31]. It is noted that IN, being a globally accepted standard, is keenly sought after as a bridging technology between heterogeneous networks to support UPT and is going through continuous developments.

In UPT, a user is mapped to a device whereas in this thesis a user is mapped to a logical workspace inside a physical location that may correspond to one or more workspaces, containing one or more network resources. Further, here a user is not assigned a universal personal number as opposed to UPT. With regard to IN, personal mobility - that extends UPT concepts - requires major enhancements in it. Since, the service logic and the data are statically attached to a single SCP in the network. The

broader perspective of nomadic computing has offered more liberties to look into needs of mobile users than UPT could offer. Finally, the approach adopted here relies on a more flexible IP network to demonstrate the concepts developed in this thesis.

3.3.2 DUET - Distributed User-Assistant for Easy Telecommunications

The authors in [32] have proposed a network architecture consisting of distributed personal agents (PA); PAs form a logically independent service control network on top of physical networks – the architecture is called DUET. In DUET, a personal agent is designated for each user that manages the user's profile and, based on it, provides personal communication services by integrating different modalities. Users can request communication services through PAs without being aware of network configuration and location of hardware resources (telephone, fax, etc). These PAs interact with objects in distributed object pool (DOP), where each object in the pool logically defines and manages a physical resource in the network, e.g., phone, mobile, fax pager, PDA, personal computer, text-to-speech converter, voice response unit, etc. Objects are distributed in multiple DOPs; they are identified using distinguished names (DNs) in a single OSI-like directory.

Unlike the approach in this thesis, DUET relies on communication through a unique personal ID only and employs automatic user registration using active badge technology. DUET supports both terminal and personal mobility, whereas in this research only personal mobility is addressed. DUET does not contain a description of what data is required to handle mobility; though, DOP in DUET is aimed to manage the data related to resources in the network - it lacks adequate design details. Their description of PAs,

which has functionality of messaging applications, is merely of an introductory nature, and thus does not sufficiently reflect the complexity of messaging applications.

3.3.3 PCSS - Personal Communication Support System

The work in [33] also deals with personal mobility and its ramifications (messaging services, which they categorize as service personalization and service interoperability). Their design approach draws on the important developments in the area of IN, UPT and mobile computing and on integration principles of TMN. And their implementation is based on X.500/X.700, Electronic location Systems and Telecommunication Information Network Architecture (TINA) -C.

Despite the similarities between the work reported in this thesis and PCSS, there are some important differences: Unlike the approach adopted here, PCSS is not based incremental approach; it relies on a universal personal number. The PCSS mobility-related data concerns have not identified site-related data, which is viewed here as an essential aspect of mobility data management. In PCSS, no consideration is given to the inter-site negotiation for building a profile of a visiting user at a new location. The approach in this work, furthermore, is distinct because of its deliberations on the adaptability of traditional LMT*s* to manage personal mobility.

3.4 Conclusion

It is found through the above discussions that there are varied approaches discovered to handle the problem of seamless communications in a personal mobility environment. Personal mobility, however, is so demanding and interesting problem that

each approach has been able to identify and contribute to functional and/or implementation aspects of it's solution.

Chapter 4

4 Architecture of Nomad's Personal Access System

This chapter presents detailed analysis and sets some design objectives for the software system that provides personal mobility support to mobile users, targeting their communications needs. Appropriately, the system is named as *Nomad's Personal Access System* (*NPAS*), because it is aimed to provide "personalized" communication access to the nomadic users - who may be surrounded by one or more fixed and/or mobile devices at any location.

4.1 Introduction

The objective is to provide anytime, anywhere, customizable, personalized communication environment to nomadic users – which is also the ultimate goal for the future integrated networks (e.g., IMT-2000). It translates into several requirements that should be fulfilled by the network, at the switching (where call control is done) layer and/or at the service or intelligent layer (which is equivalent to the IN of traditional PSTN/ISDN). The goal remains a high flying one, both for research and standardizing community. Therefore, it is interesting to highlight those constituent elements/requirements of the goal that makes it challenging, which are as follows:

- Messages generated on one network or a continent (e.g., PSTN/NorthAmerica) should be able to travel into another network or a continent (e.g., Internet/Europe), where heterogeneity of networks appears as a hindrance. It may not be much of a challenge to overcome; however, due to the scale of the problem it is not a simple one

to solve. Moreover, the assumption is that a solution should cause minimal impact on the existing infrastructure.

- Networks require an ability to address users. Since, it is users, who should get the messages not their devices in office or at home. For one, it may be addressed by providing users with smart cards. This implies that users are assigned with a unique personal address/number. The user then registers at a device to inform the network their current association to device. However, this incurs the replacement of all those existing devices that are not provided with the smart cards readers. Secondly, voice-prompted registration could be used. Since, it is a new feature for existing networks; it would require changes - not at the user end, but - inside the network. In addition, introducing the personal numbering - number portability marks efforts in this direction - in the existing networks is not a simple task in itself.

- Keeping track of users in a global network without asking the user to register through a networked device is also a tedious task. However, somewhat, it is similar to the location management of mobile phones in cellular systems. The cellular network finds the current cell of the device when the device enters a new cell and insures that the mobile device is a subscribed one when it roams into networks of independent operators. Here, a user can be located equipping them with a wearable wireless transmitter, e.g., Active Badge. Implicitly, it requires a unified or an integrated wireless system at a global level. What makes it more challenging is predicting user mobility – still an open issue.

- Once the network knows where the user is, it has to find what device(s) is in their proximity. Since, the incoming message would be transmitted to a device eventually, given that the network contains the complete knowledge on the user's wants.

- It leads to the problem of exploded communication, since then the user would receive every message. It calls for another measure: intelligent message screening mechanism that is awareness of the requirements of users. Some of these requirements may be known in advance; however, others would vary with the day-to-day activities of a user.

- On screening a message, it may be found that the message needs urgent delivery and it has to be sent in another media, since the current device association does not support the incoming message form/media. Therefore, media conversion is sought as a solution. Considering, the message as a video, screening and filtering is a challenge and then conversion to text or voice of the extracted video message, which is at least not a trivial problem.

- Quality of service supported by heterogeneous networks, fulfilling the user's needs as an alien in other's networks, security of the hosting network, and billing all combine to represent some of the issues that are still unresolved.

Thus, achieving seamless communications is apparently only an application though, but it is one of the most complex applications requiring a multidisciplinary endeavor (wireline networks, wireless networks, multimedia, intelligence, etc.). As huge a problem area, it is necessary to ask a few questions: (i) what practical assumptions can be made to simplify the scale of the problem? (ii) how to approach this problem in a way

to devise some new essential functionality to its solution? and (iii) what means should be adopted to make the new solution an incremental one? Answers to these questions are not all ideally gratifying. However, in order to generate novel network functionality certain compromises were deemed necessary, as discussed in the following section.

4.2 Analysis, Assumptions and Goal for Nomad's Personal Access System

In the context of the broad scope of the problem at hand, first it is necessary to relax some of the requirements of the ultimate goal, which were stated in the previous section. Some of those requirements/constituent elements should not be repeated in this investigation, for they are also being actively pursued in the research and/or standardizing communities. It provides an opportunity to attend to such issues that are not in limelight or are still to be discovered. Now, in response to the three questions raised in the previous section, the following deliberations are submitted:

- Considering that global scale is not possible to accomplish through a singular private venture, a smaller region should be created to test the notions developed in this investigation.

- Interoperability among all the existing and future networks remains an active research area. Therefore, it is wiser to build the testbed with the network interworking gateways (Computer Telephony Integration, Vocal Gateways) that are already available.

- Personal numbering (number portability – One Number for everything) again has received a great deal of attention and thus, it is not a young area. Further, as pointed

out in the third question, the measures adopted in this investigation should be incremental. Whereas the assignment of personal numbers has a substantial impact on underlying network. Therefore, during the transition from the current state of networks to future integrated global network, some alternative means should be provided to achieve the functionality, which is derived from personal numbering.

- Using wearable wireless transmitter allows keeping track of mobile users. Besides, several other methods could be used to find the current location of a user. Although, those means may not be as ideally convenient (from user perspective) and accurate (from network perspective) as a wearable wireless transmitter. Here, a simple means should be devised to perform location registration to determine current location of the user. Such means should be easy to implement and to integrate with the user environment, and could be relied upon as an attractive method in presence of other means of user location registration.

- Once the user is located, discovering the devices available in their proximity, and appropriately associating them to the user is an interesting research area. Therefore, it should be addressed. It promises service mobility - an intrinsic feature of personal mobility.

- Although, media screening, media extraction, and media conversion are the areas that fall in the active research domain but these are still young. Further, quality of service, security of the hosting environment, and billing, are also included in the scope of *NPAS*.

The design and development of *NPAS* is considered as a major part of the larger project of *Mobile Agent's Alliance* (*MAA*), which is a collaborative effort among, including the University of Ottawa, the National Research Council of Canada and Mitel Corporation. The goal remains that personal mobility, with communication as a target component, should be supported. However, in the light of assumptions drawn above, the initial demonstration prototype of *NPAS* will be a virtual network of the three-sites: the University of Ottawa, the National Research Council and Mitel Corp. In the following, the summary of the type of the environment, that *NPAS* would create to fulfill the communications needs of the users roaming in the three-site network, is presented:

1. To allow personal mobility, implies that devices are associated dynamically to users when they visit a new location. Therefore, the *NPAS* has to discover what devices exist in that location.

2. As described in the section 3.2, the intelligent forwarding via home of the called party allows deciding where the user is currently located. And, it relies on the information contained in the user profile that contains information on the type of devices and quality of service that the user is assigned to at the current location.

3. Despite the fact that a caller has dialed a phone number of the receiving party, the message may arrive at any other device or in any form (fax, e-mail, etc) at receiver's end. This can be described as an integrated service environment, and creating a profile for the user at home and visited location achieves it.

4. Further, the locations associated to different sites exist under different organizations; therefore, the services available to the users would be impacted by these

organizational policies. Therefore, aim is to respect these policies and yet find a middle ground to support user's communication requirements.

5. Reachability of the users should not overload user's information pool. Therefore, screening, filtering, media conversion would be in place to facilitate, what information, at what time, and in what form should be delivered to the users.

With this view of capabilities referred to as the goal and the above assumptions, the architecture of *NPAS* is described in the next section.

4.3 Architecture of Nomad's Personal Access System

This work provides a practical insight of personal mobility through the design and implementation of *Nomad's Personal Access System* (*NPAS*). *NPAS*, a software system at OSI application layer, emulates a personal mobility environment within a virtual network that spans multiple organizational networks, namely: the University of Ottawa, the National Research Council of Canada and Mitel Corporation. *NPAS* is aimed at the communications needs of mobile users roaming through the 3-site network, illustrated in the figure 4.1. Internet is used as an overlay network connecting the heterogeneous networks at the three sites. Since, Internet has seen unparalleled infiltration into corporate and public networks, and more importantly, because of its availability at the network edge, i.e., at end-user level.

The 3-site network in figure 4.1 serves as a testbed for personal mobility. It would facilitate the exploration of the practical issues concerning personal mobility. Figure 4.1 shows the three *Enabled Sites* (*ES*). The *Enabled Site* refers to the site where *NPAS* is available. A group of *Enabled Sites* forms an *Enabled Region* (*ER*). In contrast, an *Un-*

Enabled Site (*US*) refers to the site where *NPAS* is not available. Similarly, a group of *Un-Enabled Sites* forms an *Un-Enabled Region* (*UR*). It should be noted in figure 4.1 that the three agents are posted to form a virtual network among the sites. These agents are representative of proxy [34] or interconnection agents that support the mobility of subscribers roaming through heterogeneous environments at these sites.

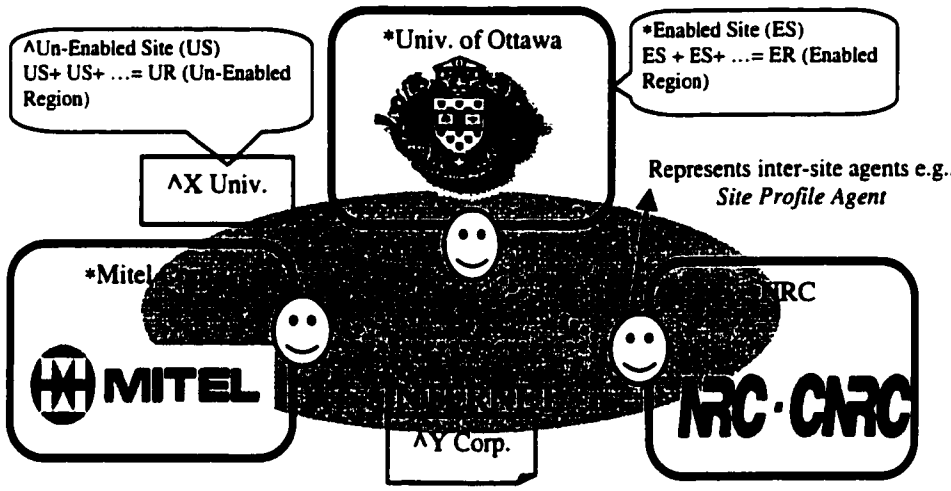


Figure 4.1: 3-Site Testbed for NPAS

The 3-site *NPAS* testbed is not designed on the principles of Virtual Private Network (VPN), which has its origin in PSTN/ISDN. VPN is built on IN architecture as a composite service application for business and corporate customers with fixed terminals and PBX. It provides a protected imaginary network inside a public network that acts like a PBX. Three features characterize VPN: private numbering plan, call screening and customized billing [35]. In contrast, *NPAS* does not use private numbering plan. Furthermore, *NPAS* not only provides communication based on fixed devices, but it also creates user-centric integrated service environment, which may be composed of one or more fixed and/or mobile devices.

Within the *NPAS* testbed, a subscriber is allowed to use shared device(s) in a new location, while away from his/her home or office desktop, fixed phone, or other personal-use portable or mobile device. This capability leads to another one, i.e. personalization of network services in a visited location. Thus, a new location becomes a 'surrogate or virtual home' for a nomadic user. It means that *NPAS* would provide *Messaging Services* in the virtual home. Such as, call forwarding with dynamically changing input from the user profile, call origination at the visited site with user's desired tariff plan, call barring for unimportant one, call notification, etc. Primitive forms of these services, without mobility support, are achieved through IN in the wired networks. As indicated earlier, *NPAS* promises to integrate disparate devices to form a user-centric, integrated service environment. Further, it seeks to cater to the user's requirements in a transparent manner.

System capabilities of *NPAS* that includes discovering the devices available in changing locations, knowing the user's service profile (user's preferences, permission and restrictions at home and visited sites), and accessing the organizational polices, call for the deployment of a persistent *Data Repository*. The *Data Repository* represents data management aspect of *NPAS*. In this work, it is considered as the basis for supporting the *Messaging Services* that are sought as ultimate goal in personal mobility environment. *Messaging Services* are applications that address the communication needs of mobile users. This distribution of the functionality, drawn from personal mobility, into two major components: *Data Repository* and *Messaging Services*, provides a rational boundary for the work that is focus in this thesis: mobility data management. In this thesis, the goal is to design and develop the components and mechanisms of data management that are required to support *Messaging Services*. Nonetheless, through *NPAS* design, it is sought

to present a simplified model to comprehend the issues of personal mobility, including (i) how does dynamic association of a user to devices offer personal mobility? (ii) what kind of data is required to achieve dynamic association of a users to devices? and also (iii) what messaging services are provided on the basis of this data? Although messaging is the ultimate goal of *NPAS*, it remains in a high level view during this thesis. Since, understanding what *Messaging Services* are needed for a mobile users, enables defining what data should be put into data management component of *NPAS*.

Examples of the mobility-related data that is stored in the *Data Repository*, includes user identity, location of the users, type of devices available in the locations, classes of services attached to these devices, user privileges, destination site policies, etc. The certain functions of data management are addressed using the agent concept. For example, agents referred to here as *Site Profile Agents* perform inter-site negotiation for the user's service profiles when users visit new locations. Messaging is largely viewed as the interaction of autonomous programs (agents) [36], representing users, services, and data resources.

Figure 4.2 shows the architecture of *NPAS*. The components are: (i) the *Data Repository* (*DR*), which deals with the collection and maintenance of data for nomads and devices available in a workspace; (ii) the *Messaging Services* (*MS*), uses *DR* information to provide messaging services to nomads; (iii) the *Intermediate Objects* (*IOs*) acts as a middle-tier between the *Data Repository* and *Messaging Services* and also serves as application framework for *Messaging Services*; (iv) the *DR Builder*, which comprises of entities that collect and create dynamic data to populate the *Data Repository*. In this work, it is strongly recognized that *Messaging Services* cannot be provided unless *Data*

Repository is in place. Since, *Data Repository* and *Data Repository Builder* are semantically related; therefore, they are grouped as mobility data management. The components of *NPAS* are elaborated on below in the following order:

1. Data Repository
2. Data Repository Builder
3. Messaging Services
4. Intermediate Objects

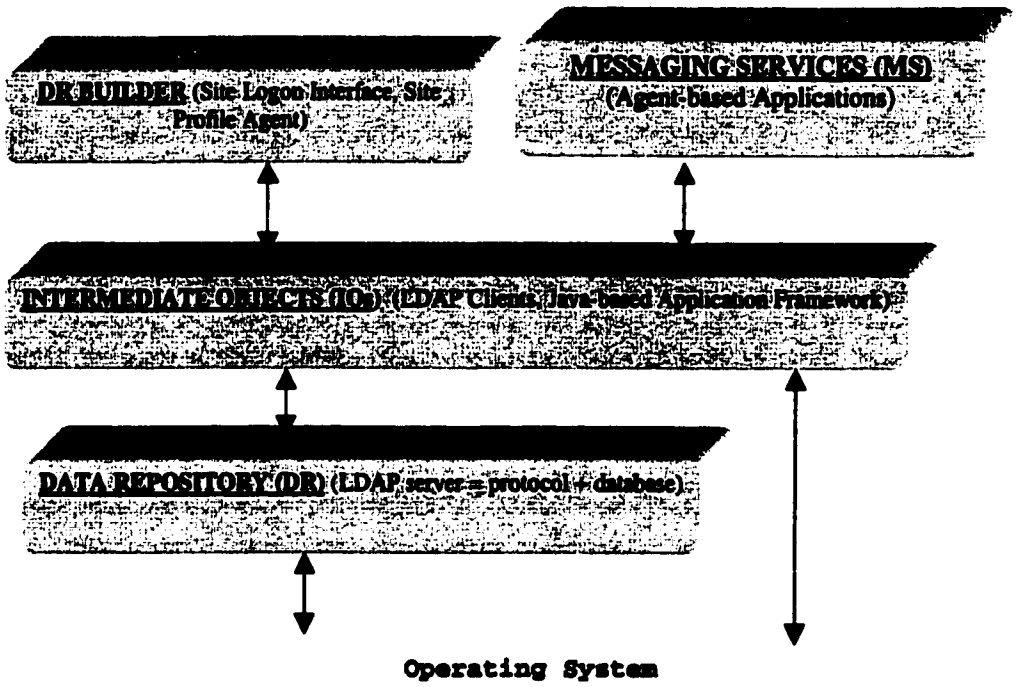


Figure 4.2: Architecture of NPAS

It is noted that security is an important issue for all the components of *NPAS*. However, this thesis only identifies its necessity to *DR Builder* (*Site Logon Interface* and *Site Profile Agent*). Due to its grave importance, the design and implementation of security features is deemed beyond the scope of this thesis.

4.4 Data Repository

In this section, first, the functionality of *Data Repository* is presented. Then, based on the functionality, types of mobility-related data are derived. Later, location management technique (LMT) of *NPAS* is discussed, which is compared with the LMT used in GSM for terminal mobility. Following that, LDAP (Lightweight Directory Access Protocol) – a directory service - is proposed to create a distributed (a partitioned) data repository for *NPAS*.

4.4.1 Dynamic Mapping of User to Devices - A Functional Description of Data Repository

According to the characteristics of nomadic environment, a nomad while in transit is exposed to varying communication and computing devices. The variety of devices is the function of the location. The current state of technology fails to provide location transparency: since, as the nomadic user leaves their home environment, their communication capability is severed. Secondly, currently devices work in a discrete manner, as stated earlier. However, a user seeks to have a self-centered integrated service environment. To address these requirements of independence of location and device, the network environment must adopt a procedure to maintain and update the changing locations of nomads and the corresponding devices associations performed for those locations. Thus, *Data Repository (DR),* the first component of *NPAS,* provides the functionality of *dynamic mapping* of users to devices.

As a first step, it requires liberating roaming users from current static associations with devices. This implies assigning each nomad a globally unique identifier, a concept that is analogous to a UPT number. However, as noted earlier, the approach here is

different from UPT. In the perspective of the incremental approach adopted here, e-mail address is selected for identifying users across organizational networks.

Subsequently, every nomad must either notify the network through an explicit logon or an implicit logon. The term logon means the same as user registration; the choice of this term identifies with the implementation done in this work. The examples of an explicit logon are typical textual logon, audio logon (user calls to inform the system of his/her current location), access-card logon (using the Smart Card as in GSM [3]), and electronic calendars of [36]. An implicit logon is to register a nomad without conscious effort on the part of the user to inform the network about their whereabouts, thereby the network, through its sensors, traces a nomad's location. The examples of implicit logon are polling nomad's terminals for user activity, employing voice authentication for out-bound calls, using an active badge/facial recognition system in a workspace, predicting location based on mobility patterns.

Although, the discussion of diverse logon (registration) modes is beyond the scope of this work, briefly some deliberations are in order. In this work, explicit textual logon is selected due to its simplicity and easier integration into the existing user environment. However, a better choice would be the use of active badge or facial recognition system, subject to the economics of the project. Nonetheless, generally, it is more likely that future systems would employ implicit logon techniques - due to the real-time location updates - as a primary means to locate a nomad. And, explicit logon would be used in places where electronic location infrastructure is not in place or in case of its operational failure. Thus, implicit logon techniques may be used in conjunction with the

explicit ones, where former serves as a primary and later as a secondary location-tracking tool.

After a logon at a new location, the hosting environment in view of the visitor needs creates a dynamic but passive association between a visiting user and shared devices available in the visited location. The passive association refers to a binding that is evaluated either at the reception of a message on the basis of the media and/or semantics of the message or at the origination of a call through device-based user authentication. This is called here as a first dynamic association or mapping. It is performed without consideration to real-time restrictions of 0.5 – 2s that is observed in traditional wired/wireless network for a call setup (for calls that require database transactions). The reasons are: firstly, the calls that are screened as urgent cannot be delivered to the called party even if the profile creation were completed in real-time. Since, the user would be on the way to a new location. Secondly, in the case when the user is already at a new site, but they have not logged on, the incoming calls that are not urgent deliverables are appropriately treated with the most up-to-date information available in the user profile at home. Finally, in the case, when a call is found as an urgent deliverable, it is assumed it can be held at the home end of the called party, and as soon as mapping is done, that call can be transferred. In the final case, momentary loss of communication ability is evident, which may be critical to certain user. However, if such a user carries a mobile device, it can rescue that call. Besides user's needs, this mapping takes into account the integrity and security of the hosting network, an important issue in *NPAS* data management. The mapping creates a 'virtual home' or a 'guest profile' for a nomad at the new location. For which, a distinctive element of data management, named here as *Site Profile Agent*, is

used. The *Site Profile Agents*, posted at home and visited sites, perform inter-site negotiation for creating 'guest profile' when a user visits a new location.

At the time of an incoming call, the network does another dynamic mapping. However, this time between a user and the ultimate device, conforming to the media and/or semantics of the message and the user's guest profile, created during the first dynamic mapping. This mapping is treated in time critical manner, if an incoming message is classified as an urgent deliverable and is of synchronous nature. In contrast to the first passive mapping, this mapping could be qualified as an active one, since it is performed to 'deliver' a message. Further, it should also be observed that unlike the first dynamic mapping, an integral step in data management, the second dynamic mapping is performed for a *Messaging Service*.

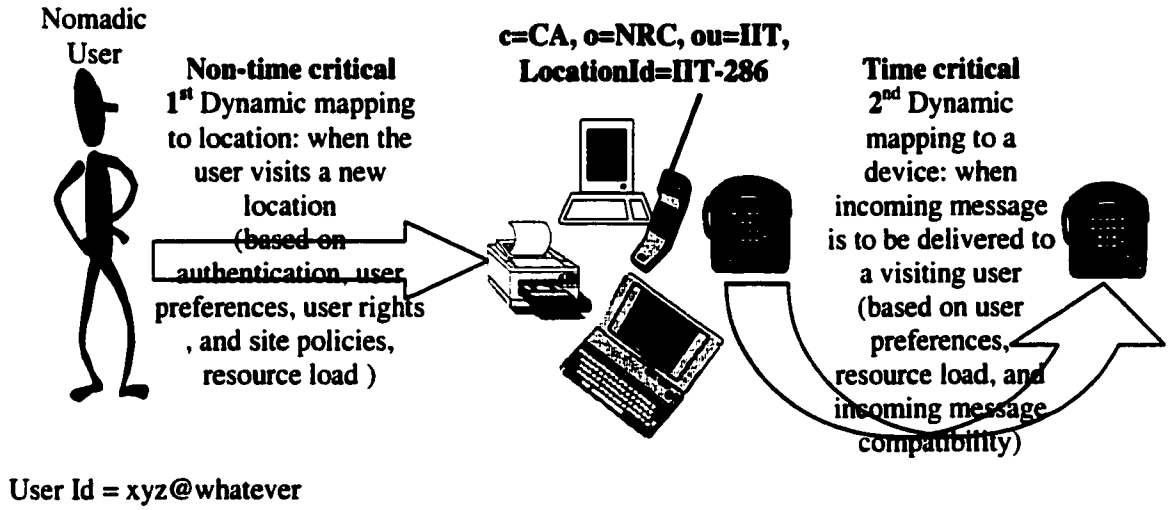


Figure 4.3: Two-Step Dynamic Mapping of User to Devices

How does *NPAS* discovers what resources or devices are available at a certain location is also a function that is handled through the *Data Repository*. The *Data*

Repository can provide the parameters/information defining the communication resources available in any location. However, it is important to note that a physical location may happen to be big room accommodating several people. A device mapped to a visiting user in such location has to be in proximity of the user, to emulate convenience available at their home site. Therefore, in *NPAS*, a user during the first dynamic mapping is associated to logical location called Workspace Area, which may correspond to the typical individual's office. The smaller the Workspace Area, the higher is the granularity of service provisioning. Since, it impacts upon the accessibility to devices from the nomad's actual position in the Workspace Area. Therefore, defining the dimensions of a logical location area, analogous to a cell in a wireless environment, is an important factor in the provisioning of personal mobility. A network site would generally contain more than one workspace. Figure 4.3 shows the example of a variety of devices available for visiting users in the Workspace Area IIT-286 at NRC, Canada.

Dynamically creating a logical workspace in view of changing locations, and providing such infrastructure data to *Messaging Services* for making appropriate call delivery decision offers a user-centric integrated service environment. The use of such data is demonstrated through examples in the section 4.6.

It can be noted that the data stored in the *Data Repository* can be either of static or dynamic nature. The static data is provided to the system in advance of starting its operation, and is long-lived. Examples of which is the subscriber identity, devices in a personal workspace at the home site, privileges at home, messaging requirements and preferences, etc. On the other hand, the dynamic data is created at run time, and is short-

lived. Such as current location of a user, type of devices and their addresses at visited site.

4.4.2 Types of Data Stored in Data Repository

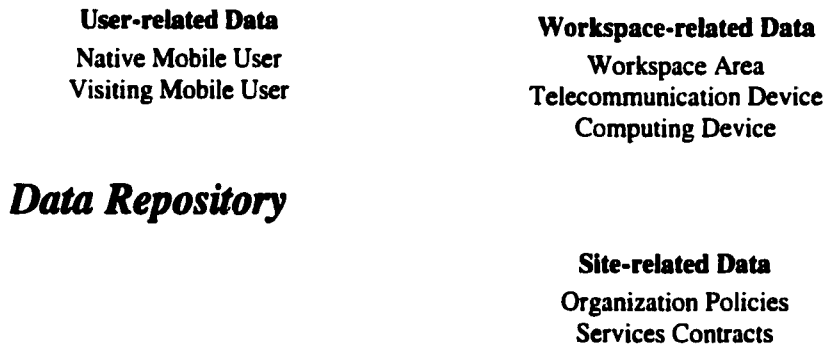


Figure 4.4: Types of Mobility-related Data

Now, based on the functionality described above, the mobility-related data can be derived. The mobility-related data required to achieve the functionality are categorized into three types, which are: (i) *User-related* data, which refers to the profile (the static data and learned data on user's location, preferences and messaging requirements) of the users at home and visited location; (ii) *Workspace-related*, which defines the types of devices that are available for visiting users in a visited workspace (location), and it also defines schema to store information on usage-load and health of the devices in that location area; and (iii) *Site-related* data, which contains information about the overall organization policies and the service contracts that are applicable to its known and new visitors.

Figure 4.4 illustrates the types of the mobility-related data, along with the physical and logical entities involved, whose profile are grouped in one of the categories. There are two physical entities, *Native Mobile User* and *Visiting Mobile User* whose data fall in *User* category. Secondly, in *Workspace* category, there are number of entities, for instance, logical Workspace Area, Telecommunication Device, Computinng Device, etc. Finally, *Site* category contain data on logical elements of a physical organization, for example, Organization Policies, Service Contracts. The details on the attributes of these types of data are given in the next chapter.

4.4.3 Location Management Technique of NPAS

There should be an algorithm defining the data building mechanism commencing from profile creation for subscription at home to the time when the profile update occurs as user changes location, where one step would be the first dynamic mapping. The algorithm is referred to here as location management technique (LMT) - in line with the term used for terminal mobility in wireless systems, where the objective is to find the location of a device. LMT in a generic sense is a tracking mechanism that essentially involves two operations: move and find. In a move operation, a system updates the location of the user to the new address that identifies their current location after a move occurs. In a find operation, system performs a search to determine the current location of a user [37].

The LMT of *NPAS* is applicable to *Intra-Enabled Region* movements, i.e., for a user who roams between the *Enabled* sites. The objective of *NPAS's* LMT is to define a step by step process for building data in *Data Repository*. Thereby, this would allow finding the current location of a roaming user and their devices associations in a new

location. In figure 4.5 a UML (Unified Modeling Language [38]) Activity diagram depicts the steps in the LMT of *NPAS*.

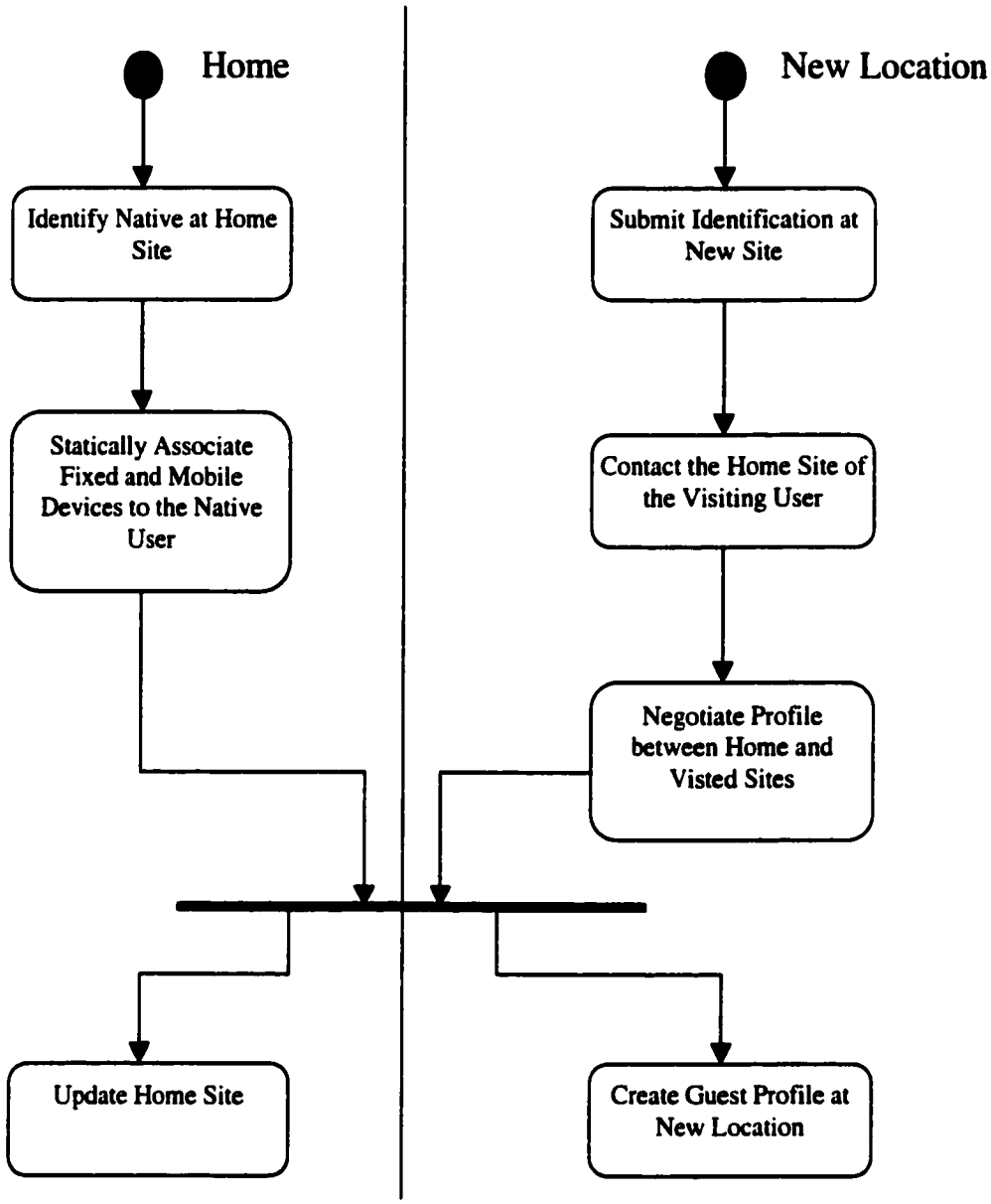


Figure 4.5: UML's Activity Diagram for LMT of NPAS

First, every subscriber should have a home site, which maintains data on its *Native Mobile Users* (*nmUser*), which in a semantic sense is the 'home profile' of a user.

The subscription at home is performed using a home e-mail address that remains valid until the user changes their home. The subscription involves the static, passive association of the user to devices available in a Workspace Area at home. These activities: to identify natives and to create static association, are performed at the home site of a user, as shown in figure 4.5. What happens at new location? Since, the user has subscribed to personal mobility services, when the user visits a new location, they must inform *NPAS* of the new location. It is depicted in the figure on the right side. At a new location, the *nmUser* becomes a guest and is represented as the *Visiting Mobile User (vmUser).* The new location also identifies a *vmUser* by their home e-mail address. Therefore, a *vmUser* submits their home e-mail address along with their identity authentication through a *Site Logon Interface*, a textual logon interface.

Based on the data, minimum one and maximum all enabled sites may be searched to authenticate the user's home. If the search is successful, negotiation is performed between the visited and home sites to create a 'guest profile', which represents a *vmUser.* The negotiation is held between the *Site Profile Agents* at home and visited sites. These agents dynamically assign devices (fixed and mobile) based on the several criteria, described later. As a final step, the negotiated guest profile is sent to the home site and a copy is saved at the visited site. This is shown in the figure as a concurrent activity using synchronization bar notation of UML Activity diagram. If the result of the authentication search is negative, the visiting user does not exist in the any of the data repository, and it is assumed that the user is not a subscriber of *NPAS*. Therefore, the logon is refused. The *NPAS's* LMT primarily traces the home of a *vmUser* and updates data at the both ends after the *Site Profile Agents'* negotiation.

The *NPAS's* LMT follows a "partial-information" strategy. According to which, not all vertices/nodes in a graph/system are aware of the current location of a user after they move [37]. Further, it is noted that *NPAS's* LMT is essentially the same as Home Location Register (HLR)/ Visitor Location Register (VLR) scheme that has been in use in IS-41 and GSM [39]. The similar scheme underlies the Mobile-IP (to support terminal mobility), mentioned in section 2.2. The reason is the HLR/VLR scheme, in its essence, is a natural scheme and thus, it is so generic that it addresses mobility in various domains. Due to the similarity to the HLR/VLR scheme, *NPAS's* LMT inherits its weaknesses and strengths. However, the difference is observed in terms of the amount of static and dynamic data that is involved in creating and maintaining *Data Repository*, particularly, the dynamic data that represents the creation of a new *vmUser* entry - this is not true of traditional LMTs [40].

A historically known problem of the HLR/VLR is that it produces significant network traffic, which is true for *NPAS's* LMT, as well. This problem is addressed in [37], [39] by tracing mobile users through forwarding pointers to identify their trails. However, it cannot be adopted directly in *NPAS's* LMT. Since in *NPAS's* LMT, there is a need to communicate with the home site before creating the 'guest profile' at a new location. This highlights LMT differences between personal mobility and terminal mobility. Another known demerit of the HLR/VLR scheme is its routing redundancy, which is addressed by using hierarchical databases [24]. This solution also holds promise for personal mobility. *NPAS's* LMT is based on e-mail addressing, which is tied to organizational and/or geographic boundaries. Therefore, when a user wishes to settle in a new location (organization), they are forced to get a new e-mail address. This implies

restricting mobility. It is similar to the problem of location-dependent numbering in terminal mobility [39][41].

4.4.4 X.500-Based Database Architecture

NPAS does not deploy wireless database architecture e.g., IS-41 in USA, or GSM in Europe for a 3-site testbed. In those cases, a signaling protocol between the nodes of the network (Mobile Switching Center – MSC, HLR, VLR) is based on Signal System 7 (SS7). Example of such a protocol in GSM and IS-41 is Mobile Application Part (differs in Europe and US) that talks among network elements (MSC, HLR, VLR). Implementing or getting access to such a protocol stack, designed for a public network, is overkill for a private network. Similar to HLR/VLR in wireless database architecture, X.500 Directory Service in Open Systems Interconnect (OSI) networks provides an attractive solution for creating a distributed database. X.500 is also considered by others in building private telecommunication networks [43][44]. However, their goals differ from *NPAS* - where their aim is to enhance PABX for supporting mobile terminals (Cordless Telephone Mobility – CTM) for a wide area or/and the provision of personal mobility support in private networks based on UPT approach.

A directory service is a name binding and resolution service. However, it is unlike a name service (white pages): an application that allows a database lookup for attributes from a given name. In contrast to a name service, a directory service (yellow pages) is designed to support attribute-based search; i.e. both name and attribute can be used to lookup records in the directory database. The *Data Repository* will be queried in situations where an attribute-based search is required. For instance, when searching through a fixed-phone number to find a user's fax number. Therefore, directory service

lookups or query can be compared to call setup and also choosing a device from a user-centric, integrated service environment (workspace) is a call switching and call control activity.

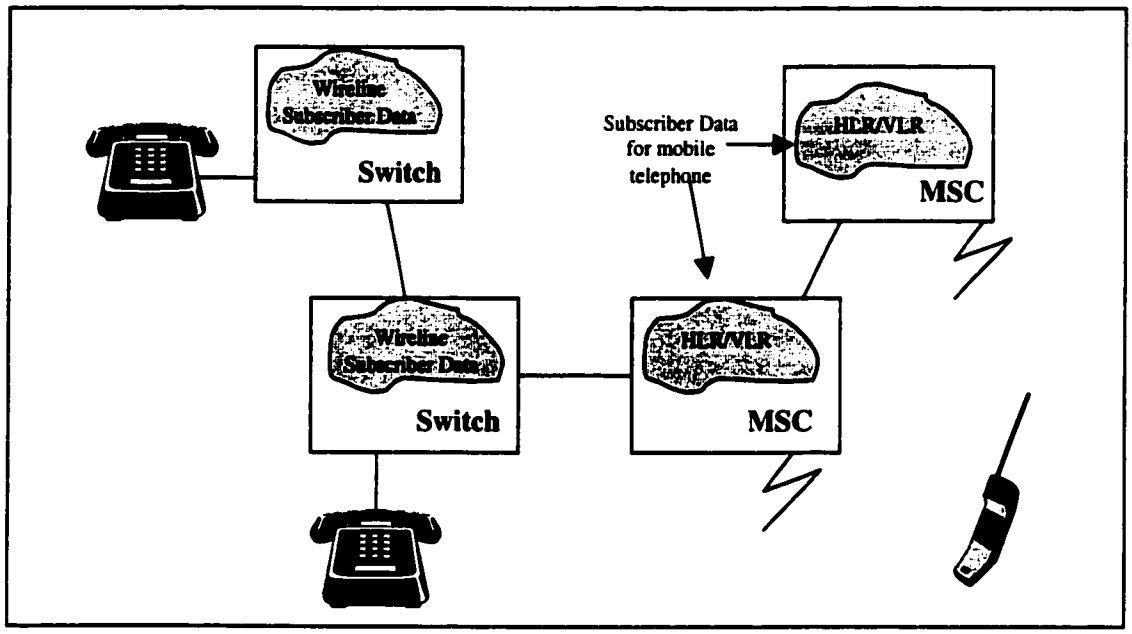


Figure 4.6: Existing Wireline and Wireless Networks

X.500 is the International Standards Organization (ISO) standard for directory service [45]; and, it is implemented on OSI protocol stack. The ISO's X.500 distributed directory standard through stand-alone LDAP Server (lightweight version of X.500) has recently been deployed at a massive scale for a local (Intranet) use. The Lightweight Directory Access Protocol (LDAP), the IETF directory standard [46] for a TCP/IP network, provides a directory service that is (emerging) distributed, scalable and highly protected. Therefore, LDAP is selected to implement the distributed *Data Repository*. LDAP increasingly provides almost all the ingredients of X.500; the distributed model is not fully-fledged as of LDAP v3 [47], though.

The following main aspects of the LDAP directory are referred to as models. The four models cater to the known requirements of a distributed directory service. These models are: (i) Information model, which specifies the type of information that can be stored in the directory, i.e., to identify an arbitrary entity (a real-world entity, concept, etc); and what syntactic requirements should be complied with to define an entry (a database record of an arbitrary entity) in X.500 directory; (ii) Naming model, which defines the hierarchical naming space to organize and reference information created through the information model, the naming space is called Directory Information Tree (DIT); (iii) Functional model, which provides operations (search, add, modify) that are required for accessing the Directory Information Base (DIB, a collective name for information stored in DIT); the LDAP standard has eliminated some less useful, esoteric operations of the X.500 functional model; (iv) Security model, which defines robust authentication, access control and encryption mechanisms to protect the DIB from malicious intruders. Figure 4.7 illustrates that each *Enabled Site* maintains a Directory Information Base (DIB) for its organizational network.

Although, LDAP's standard offers limited distributed support, some commercial implementations claim to meet a few of the requirements including replication and smart referrals. However, the LMT designed for *NPAS* does not require inter-server protocol like DSP (Directory Service Protocol) of X.500. Since, the search for a mobile user through *NPAS's* virtual network is performed using iterative-navigation, where a LDAP client contacts successive servers as necessary in order to resolve the query. Further, the advantages of X.500-based approach and some performance issues will be presented in next chapter.

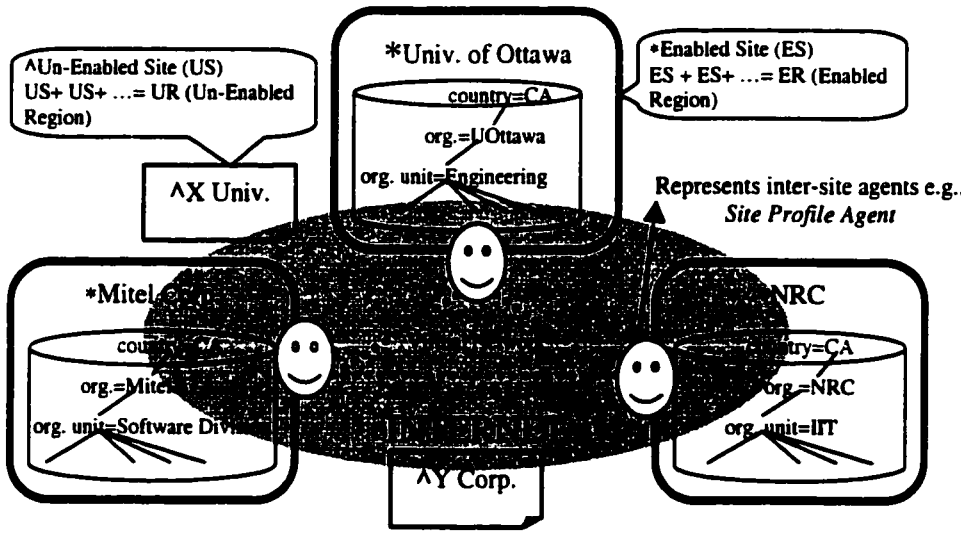


Figure 4.7: Distributed Directory for NPAS

4.5 Data Repository Builder

The complementary components that are used for mobility data management in *NPAS* along with the *Data Repository* are *Site Logon Interface* and *Site Profile Agent.* These components are described on below. Other entities contained in *DR Builder* include *Site Resource Manager*, *Site Service Gateways*, and *Device Agents*. These entities are essential to the operation of *NPAS*. However, they are highlighted here only for the sake of identification of their position in the *NPAS* architecture.

Site Resource Manager determines how to allocate, monitor, optimize and release resources available in the network of a certain site [48]. A resource also has its *Device Agent* that informs the status of a physical device to *Site Resource Manager* [36], [49]. The information available from these entities is used to populate *Workspace-* and *Site-related* data in the *Data Repository*. Further, the *Site Service Gateway* hides the resources

internal to the network of a site from those who are external to that site. In other words, it is a proxy of the network resources.

4.5.1 Site Logon Interface

As part of the incremental approach, as stated earlier, the explicit logon is chosen in preference to electronic ones. For the initial demonstration prototype of *NPAS*, .a textual logon is designed. The following assumptions guide the development of this logon interface:

- The minimum amount of data should be required for *NPAS* user to supply at the time of logon.
- The interface should be accessible from any networked computers, and therefore, it should be web-based.
- The interface should be based on thin-client concept popular for Internet applications.
- The interface must contain security features that protect against spoof.

Further details pertaining to its implementation follow in the next chapter.

4.5.2 Site Profile Agent

To dynamically map a set of devices and the services associated to those devices (at visited site) to a visiting mobile user has been called a "surrogate or virtual" home, which is the aim in the first dynamic mapping. The provisioning of communication services in a user-centric manner also depends on how closely this guest profile reflects a user's home workspace and the current needs of the user. Further, this profile cannot be

created in isolation from the visited site policies. Since, it is up to an organization to determine whether a visitor can be permitted to access their local resources. This is an important issue where a middle ground between the concerning parties, i.e., a visiting mobile user and the hosting site, must be reached. However, once a user arrives at a new location it can not be expected of them to initiate the process of informing what are their preferences, and then determining what services can be supported. Therefore, at the user-level the complexity of this step should be hidden. This presents an interesting application area for software agents, where agents act on behalf of a visiting user and visited site to handle the process of finding a middle ground for the two parties. These agents are referred to here as *Site Profile Agents*.

A primitive form of the functionality, being achieved through *Site Profile Agent*, is already known in the existing networks. In fixed telecommunication networks (includes IN-based), an operator is chosen at the time of service subscription, whereas in GSM, a mobile user enjoys a number of standard services by subscribing temporarily at any location where a cooperative network exists. Here the cooperative network is formed based on a Memorandum of Understanding (MoU), which defines willingness to facilitate transfer of subscriber information between operators. In addition, the enhanced operator specific services (IN services) are also now aimed to be provided in any GSM network, which is currently being standardized in GSM Phase 2 as Customized Applications for Mobile network Enhanced Logic (CAMEL) [50]. Further, in the work of [51], the CAMEL functionality is extended to include PSTN, along with the GSM networks. However, the aim of these works is to map the services associated to a single, personal device to different or the same device in the new network environment. Whereas

the goal of *Site Profile Agent* is to create a workspace i.e., "surrogate home" at a visited site, regardless of device (be it mobile or fixed).

4.5.2.1 Definition and Characteristics of Site Profile Agent

Site Profile Agent is an inter-site negotiating agent that communicates with its peer to mutually decide computing and communications services for a roaming user. These agents negotiate solely one-on-one. The negotiation observes the three criteria:

- Respects the policies, security, and authorizations of the visited site;
- Protects a visiting user's profile against snooping and vice versa (i.e., protects a hosting site information from a malicious agent);
- And fulfills the requirements (communications and computing resources) of a visiting user for a "virtual home" at a visited site.

Site Profile Agent only interacts on behalf of the user who is a subscriber of an *enabled site*. The sites involved in the user profile negotiation are assumed to have an a priori organization-level service-subscription contract. A service-subscription contract reflects a mutual agreement of hosting visitors, setting the premises for a co-operative network environment. This organizational-level contract endorses the willingness of a site to be a 'surrogate home' for known visitors and explicitly specifies a set of pre-defined classes of services for those visitors. The visitor-level services-subscription contract identifies a visiting user, as an individual who is not affiliated to an organization. *Site Profile Agents'* negotiation only determines a resources (or services) profile (i.e.,

resource allocation) for a visiting user. It has no influence in the establishment of any organization or visitor-level services-subscription contract.

Site Profile Agent is not able to take input or feedback from the visiting mobile user. This enhances the security of the sites involved. To perform their roles autonomously, as representatives of a visiting user and a visited location, respectively, they depend solely on the user, location and *Site-related* data stored in the corresponding *DRs*. If the negotiated profile seems unsatisfactory to the visiting mobile user or to their *User Agent* [36] [52] (it knows about nomad's messaging criteria and performs the work of a personal secretary), it is assumed it would require human intervention. Such intervention comes from the user by either modifying their home *DR* or informing the visited site's *DR* administrator.

4.5.2.2 Negotiation Between Site Profile Agents

Negotiation is referred to here as a communication activity that determines an acceptable solution for the parties that may have conflicting proposals but share a common aim. The use of negotiation by *Site Profile Agent* is natural, since both the visiting user and the hosting site generate an interaction where there is a likelihood of conflict of proposals. The following conflicts may arise between *Site Profile Agents*:

1. Requirements and preferences of user vs. security and privilege policies of hosting site

2. Requirements and preferences of user vs. current device load at a hosting site

3. User expectations vs. quality of service at hosting site

4. Requirements and preferences of user vs. billing of services at hosting site

5. Revelation of user's information vs. the information sought by hosting site and revelation of site information vs. the information sought by user's *Site Profile Agent.*

A service-subscription contract resolves the conflict no. 4. However, it is more flexible to be able to determine the charges of services at run-time, as well. Since, a new discount or savings plan may have been introduced after the establishment of service-subscription contract. In view of the sensitive information, both parties may only exchange the information that falls in the scope of the negotiation, which is the assumption for conflict no. 5. Therefore, the exchange should be precisely directed at resolving any of the conflicts only. Further, to maximize the advantage to the user and the hosting site, neither agent may reveal their minimum acceptance criteria until a certain stage of negotiation; however, this subtlety is not addressed here. Instead, to simplify the problem, it is assumed that the both parties trust each other in this respect.

Examples of services are phone, fax, e-mail, etc. These services could be further classified as either required or preferential. Here, the required service refers to a service that is essential to a user, whereas, the preferential service indicates a service that is desired but is not essential. In addition, a basic service may have enhanced levels, based on coverage and/or quality of service - a specialization of that service- e.g., domestic long distance, overseas calls, etc. A proposal or counter-proposal will have attribute as service name and its value would be a speciality level(s) and some other parameters that define the service constraints, if any (e.g., availability during certain time limits).

It is noted that the conflicts facing *Site Profile Agent* does not impact upon (add to or reduce) feature interaction problem. Although, it may seem that when devices are mapped the services/features will also be mapped. However, this mapping of devices to a user is an activity that takes into account only the resources that are needed by the applications in *Messaging Services*, which handle message inflow for a native mobile user. Therefore, feature interaction is still the focus of *Messaging Services*, since it is encountered during a messaging session.

Based on the conflicts addressed here and the definitions of elements of negotiation, now a high-level strategy for negotiation is developed. Primarily, *Site Profile Agent* conflicts are of resource conflict type [51]. The conflict occurs here, primarily, due to the interaction of the user constraints (i.e., requirements and preferences) and site constraints (i.e., policies and resource load). The parameters contained in proposals and counter proposals, defined by the user and site constraints, are known at the time of negotiation from user, workspace and *Site-related* data available in the *Data Repository*. Further, instead of exchanging a sequence of proposal and counter-proposals, a hierarchical data structure called here *Services Option Tree* is sent by the *Site Profile Agents*. It is the hosting site that first sends out the *Services Option Tree*, containing information on what type of services are available at hosting site, what are their interdependencies, etc. The interdependence here refers to the fact that if one service is chosen from a branch then other services can or cannot be chosen from that level. On processing, the user's home site returns a processed *Services Option Tree*. The incoming *Services Option Tree* contains the selected bids of proposal, i.e. the subset of the tree. The *Services Option Tree* plays two important roles in the agent negotiation. First, all possible

proposals are included in the outgoing *Services Option Tree*; thus, it overcomes the need to create counter-proposals dynamically. Subsequently, the list of choices provides the sufficient flexibility to select the desired proposals. It is assumed that outgoing *Services Option Tree* is the honest representation of services available. Secondly, the *Services Option Tree* reduces the overhead of messages and thus greatly simplifies the conversation. However, this requires that *Site Profile Agents* be complex enough to construct and process *Services Option Tree*. Nonetheless, in addition to *Services Option Tree* messages are exchanged to authenticate the agents and to start the negotiation. The sketch of the protocol is shown in figure 4.8. Details on it follow in the next chapter.

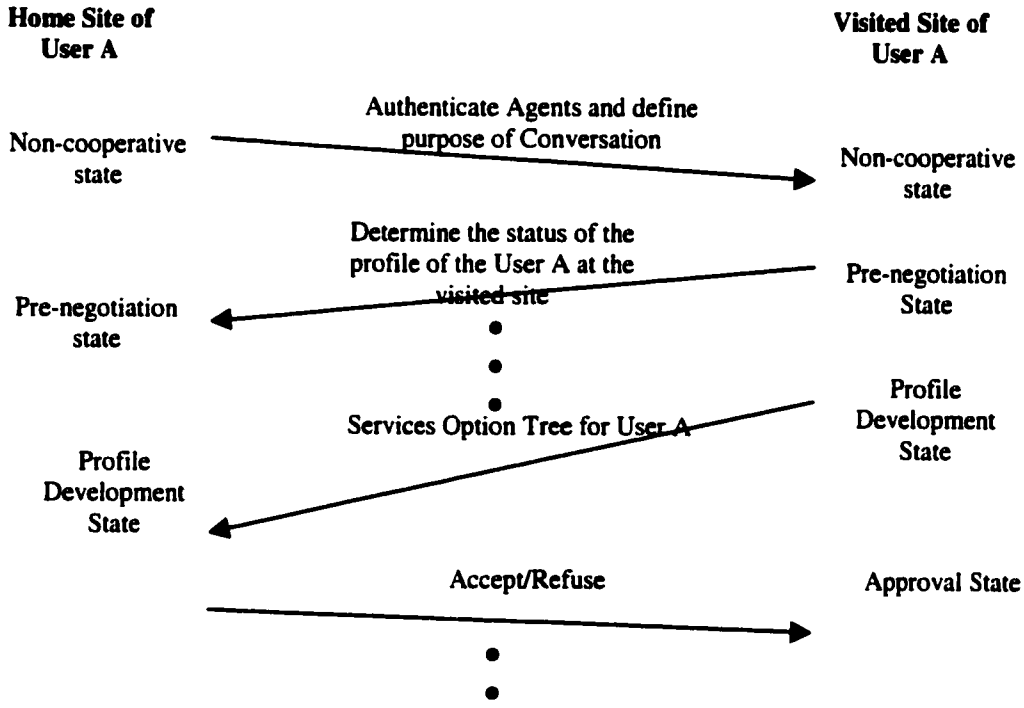


Figure 4.8: Site Profile Agent Negotiation

4.5.2.3 Inter-Agent Communication Using KQML

The realization of software attributes indicated above is by no means limited to the agent paradigm, since legacy software applications in the AI and distributed computing could already offer many of the above attributes [54]. However, it is widely accepted of 'agents' to be considered as a software program that can communicate through formalized message-based interfaces (e.g., KQML). These interfaces are application (or object) independent. Whereas, in object-oriented programming interfaces vary from one object to another. Additionally, message-based interfaces are transparent to the underlying programming language, transport protocol, and operating system. They form a proposed standard communication language for distributed multi-agent applications. This view distinguishes agents from the legacy software; however, it does not divorce KQML-speaking agents from being programmed in any programming language and reaping the strengths of a contemporary programming paradigm. Therefore, today's distributed software agents, being enhanced descendents of the legacy software, are sought to communicate through object-independent messages to achieve increased interoperability.

KQML messages are developed from the speech-act theory (a theoretical account of human communication). A KQML message contains speech-act (also called performative, with an understanding that it will result in action), which provides intention (attitude) of the message being sent. The pragmatic information (who is the sender and receiver, how to understand what is said, and how to identify a message received) and finally, an arbitrary content type, e.g., ASCII, and binary format, follow the speech act. The part of pragmatic information serves as a message-handling protocol. Here, *Site*

Profile Agents communicate using speech-acts and the message handling protocol of KQML.

KQML does not commit to any interaction protocol: a sequence of message exchanged in a multi-message conversation. However, it does provide a set of primitive conversations, conversation policies, to validate the semantics of the performatives [23]. The conversation policies empower a KQML-speaking agent to select primitive conversations. The selection is based on the performative of the first incoming message. However, it does not give the ability to carry on a valid, arbitrary conversation. In applications, where an agent at the time of its creation is known to have a specific purpose, e.g., *Site Profile Agent's* conversation, it becomes redundant to put these conversation policies for semantic validation. Instead, where necessary, agent should be programmed in a way that if a need arises, it could select from the conversations provided to it [34].

4.6 Messaging Services

The *Messaging Services (MS)*, the second component of *NPAS*, constitutes messaging applications for all forms of human communication, including voice, text, graphics, and video. The messaging applications are responsible for delivering incoming and outgoing single-media or multimedia messages by making appropriate interceptions. For instance, acquisition of the current location data on a nomad, filtering a message based on the personal profile of a nomad, conversion of media, and/or abstraction of video content. The conversion and/or abstraction for a message is requested by a *Messaging Services User Agent*, when at least one of the following constraints is

applicable: (i) a user is on a low-bandwidth connection and the message is a bandwidth-intensive one (e.g. image, video), (ii) the user does not wish to spend time and/or money for retrieving/sending a large multimedia file, (iii) the receiving device is incapable of displaying the media, due to a small display (e.g., pager, PDA), unavailability of media driver (on PC or Laptop) or incompatibility of incoming media format (e.g., a phone cannot display image or graphics on a fax). Some of these constraints may also be caused by visited site policies, which determines the use of resources in its jurisdiction. The messaging environment is sensitive to both the content as well as the media of a message. Further, Quality of Service (QoS) requirements, posed by different media, introduces a negotiation process for resource allocation that immediately precedes multimedia message delivery. The multimedia-messaging requirements bring forth a facet of negotiation, which calls for the application of software agents.

It is noted that the IN approach to feature or service development is not scalable in nomadic environment. Here, simple interaction between network and user is no more a goal, since the user desires for personalized messaging. A nomadic user is in full control of defining reachability. Thereof, a user would change their reachability through numerous ways in a dynamic manner. Moreover, feature interaction – the undesirable interaction between IN features like Call Number Delivery vs. Unlisted Number, Terminating Key Code Protection and Call Forwarding, etc. – is also sought to be resolved through agent negotiations [51], [55]. The capabilities required in such applications involve autonomous entities watching messages and triggering appropriate actions. Therefore, *Messaging Services* depend on the semantics of autonomous agents.

Messaging Services is the focus of [36] - a collaborative member of *MAA*. This work extends theirs in the following respects: the formation of a network scenario for messaging applications, identification of concrete supportive roles (to *User Agent*) to facilitate messaging, which are also attributed to agents and strategic posting of these agents in the network.

The *Messaging Services* are depicted through a network scenario in figure 4.9 below. The *Messaging Services* demonstrate the utilization of the data created in the *Data Repository*. The messaging applications are conceptualized on the assumption that messages can travel between data and telephony networks through a gateway, for instance, a Computer Telephony Integration (CTI) Server Platform. A CTI platform provides a full duplex linkage between telephone and data networks. The figure also depicts a *Unified Message Box*, which is capable of handling multimedia messages and substitutes for multiple inboxes (for voice, pager, e-mail) that exists today.

The three agents: (i) *Message Agent*, (ii) *Addressing Agent* (iii) *User Agent*, are exploited in all messaging applications. The *Addressing Agent* first finds if the called number corresponds to *NPAS* subscriber. If the result is negative, the call is handled as normally. If the result is positive, the *Message Agent* determines the type(s) of media (e.g., audio, text, image, video) a message contains. Then, the *Message Agent* provides this information to a *User Agent*. The *User Agent* based on several criteria of personalised messaging (instructed or learned) of an owning user determines that message or notification has to be delivered urgently or not. In view of the decision, the *User Agent* makes a request to the *Addressing Agent* to find the address of the compatible device that is currently accessible to the called party. The *Addressing Agent* acquires

current location and the compatible device from the home *Data Repository* of the called party. It then returns the results to the *User Agent*. If the requested device type is available, the *User Agent* conveys the device address to the gateway for completing the message delivery to the called party.

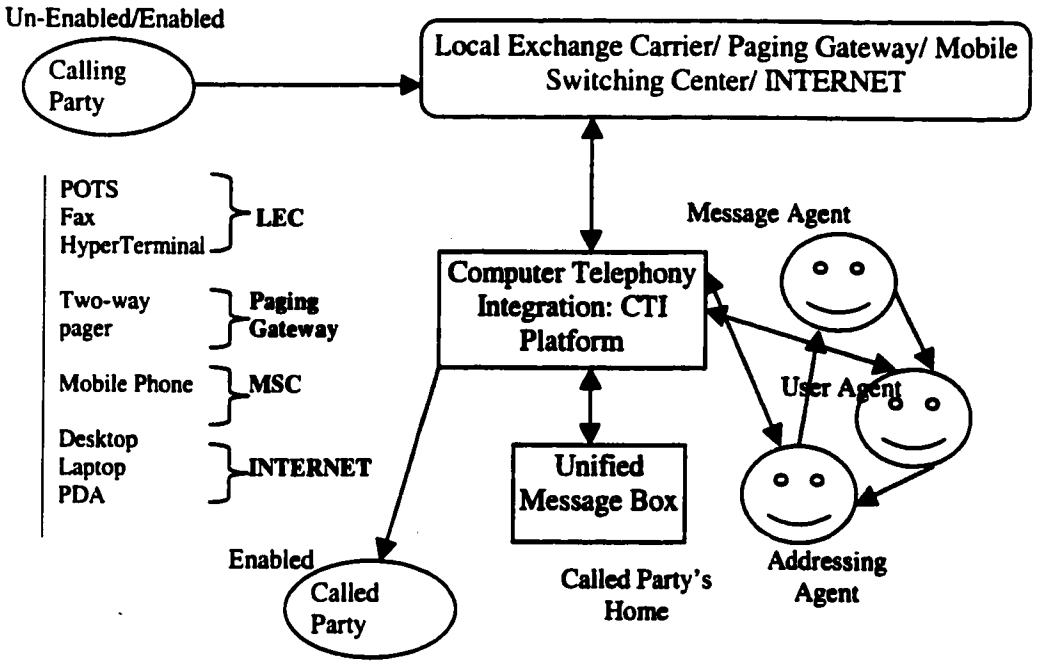


Figure 4.9: Communication from US/ES to ES Using Telecommunication/Computing Devices

4.6.1 Communication Using Telecommunication Devices

This scenario shown in figure 4.9 deals with a message originating from a fixed or mobile telecommunication device (fixed phone, mobile phone, two-way pager, HyperTerminal) in an un-enabled/enabled site and is destined for a user in an enabled site. Now, consider that a message originates from a telephone at an un-enabled site. A sender calls the known telephone number of the recipient, who is known as *nmUser* to the enabled site. How a message will go to the current location of the called party? To answer this, the *Message Agent* is posted at the CTI Platform at the home side of the

called party. In addition, it is assumed all the incoming calls at called party's home side go through the CTI Platform. Therefore, the CTI Platform is being constantly monitored by the *Message Agent*, which scans all the incoming messages. It passes the request containing the media, caller id, call's time and date to the concerning *User Agent*, which is owned by the *nmUser*. Assume that the *User Agent* finds this call is an urgent deliverable; therefore, it requests the *Addressing Agent* to get the address of a compatible device. If a device were available, the *User Agent* commands the CTI Platform to forward the call to the device available at the user's current location. In case when the *User Agent* receives the reply that a compatible device type is not available, then it can take one of the several actions, including store the message in the *Unified Message Box* and generate the *Message Notification Slip*. The *Message Notification Slip* may contain the caller's telephone number, the date, the time called and the *Unified Message Box* number for message retrieval. The *Message Notification Slip* can be sent to a recipient at an alternative device that could be e-mail, or fax. However, an alternative device may require media conversion. The *User Agent* can also provide the "Device Unavailable" message for the sender, if appropriate.

Consider another example, where a sender's source of message is fax. The message goes to recipient's home CTI Platform, as before. The strategy to find the current compatible device, which is accessible to the called party, remains the same. If the fax were not accessible, then the solution to handle the text on the fax would be to either convert the text into speech or generate a *Message Notification Slip*. However, if the message is an image only then either a *Message Notification Slip* or image abstraction service can be used to extract the content, then the result can be e-mailed to a recipient.

4.6.2 Communication Using Computing Devices

It is about the communication using single-media or multimedia message (text, graphics, voice, video) that originates from a fixed or mobile, computing device (laptop, desktop, PDA) in a un-enabled site and is destined for enabled site user. Figure 4.9 depicts the underlying network scenario.

Let's take an example: A sender is using e-mail as the means of messaging. Normally, e-mail is stored at the mail server of a user and a user can only check e-mail if he/she logs onto their home mail server. In the *NPAS* environment, it is presumed that e-mail also gets forwarded to the nomad's current enabled site. Such e-mail forwarding normally means creating a '.forward' file at the home mail account. However, in the approach here, the concerns are: (i) an e-mail account has to be dynamically created for *vmUser* at the visited location; (ii) when a *vmUser's* account life times out, the entry for *vmUser* is wiped out including e-mail account from the visited location. The first concern can be resolved once the security of visiting e-mail server is respected. The second problem implies that mails need to be transferred back to the home mail server of *vmUser* before the accDeletime (i.e., account deletion time) expires.

The network scenario in figure 4.9 can forward e-mails dynamically, if the *User Agent* gets (a) the current location of the user and (b) the mail server address from the *Addressing Agent*. In addition, either the incoming text message can be converted from text to speech or a *Message Notification Slip* can be generated. If the e-mail message has a large-video file attached to it, the video abstraction service could be used for the reasons indicated at the beginning.

4.6.3 Implications

The responsibility of the *Message Agent* requires it to scan each message even when the recipient may not be a subscriber to enabled site. Therefore, it calls for finding a solution that either avoids or at least minimizes the computation required for the scanning. Another, observation is that the scenario, in figure 4.9, differs only in Local Exchange Carrier or Paging Gateway or Mobile Switching Center, or Internet. This gives a uniform view to messaging applications, providing the basis to treat all messages equally, irrespective of whether they originate at enabled or un-enabled site.

It is important to recognize that if a recipient is always accessible from the external world then the user may get an overflow of messages. This means, it would be necessary to have a static *User Agent* [36] at the home end to mediate the message delivery decisions.

4.7 Intermediate Objects

The *Intermediate Objects (*IOs*)*, the fourth component of the *NPAS* architecture, is conceived from the implementation perspective. The first objective of *IOs* is to act as a middle-tier forming a three-tiered architecture [56], widely used in web-based form processing. It offers a common channel for all *Messaging Services* agents/applications to access the information collected in the *Data Repository*. The second objective of *IOs* is to provide an applications framework for building messaging applications.

4.7.1 Three-Tier Architecture

Three-tiered software design is the natural evolution of popular two-tier client/server architecture. Introducing an intermediary between a client and server (e.g., a

database server of a commodity) provides transparency to the client from the changes that may happen in the back–end server. This transparency is harnessed through the middle-tier/intermediary that would appropriately handle the changes at the server-end without modifying the client requests. Further, such an architecture affords scalability: a client has only one access point which is the middle-tier, whereas the middle-tier takes over the burden to make multiple access as the need arises. Due to the advantages of transparency from the changes that may occur in *Data Repository* later and the inherent need of scalability, the middle-tier architecture is selected.

4.7.2 Applications Framework

This layer would contain a software library/building blocks that provide a set of functions that are common across all applications at the level of *Data Repository Builder and Messaging Services*. Such a framework serves as a middleware software and it can be built on top of an existing middleware platform (e.g.,CORBA) that provides a higher-level communications, among distributed objects, relative to sockets. Since, the atomic functions or objects within agent-based applications should still be implemented using well-proven object-oriented techniques.

4.8 Discussion and Summary

Based on the definition of the middleware layer of ODN, it would be appropriate to include *Site Profile Agent* and *Data Repository* of *NPAS* as the middleware services. Since the services of these components is used by higher-level messaging applications of *NPAS*. Thus, the functionality set of these components indicate those elements that middleware layer of ODN could consider as a basis for supporting higher-level

messaging applications. Further, the messaging applications provide increasing interactive control to end users, which is achieved through agent negotiation protocols. These protocols could be developed using KQML. Moreover, it will be seen often that agents require some negotiation ability to perform their tasks. Therefore, it would be an interesting issue to determine what kind of horizontal negotiation services can be created at the middleware layer of ODN to support higher-level applications.

The design of *NPAS* explored the rich data required to support personal mobility based on the concept of dynamic mapping. In addition, software agent is involved when performing profile negotiation during first dynamic mapping. At various points, the analysis and design also highlighted, where necessary, the related work to provide the rational and appreciation for the investigation of *NPAS*.

Chapter 5

5 Design and Implementation

This chapter builds on the bits of design presented in previous chapter. It presents a more detailed design and implementation of those components of *NPAS* that realize mobility data management. It involves: firstly, the data model for *Data Repository*, secondly, those objects in the *Intermediate Objects* that fall in the scope of mobility data management, and finally, the *DR Builder* components: the *Site Logon Interface* and *Site Profile Agent*.

5.1 Overview of the Implementation

The figure 5.1 shows the complete view of those components that are involved in the mobility data management. A native mobile user, say User A, only sees the *Site Logon Interface* when interacting with *NPAS*; thus, it is called client side of the implementation. According to the LMT of *NPAS*, first, the user should inform the system about their location where the user is visiting. This requires the user to submit their *NPAS* identification, i.e., e-mail address and the location-related data. Thereby, the request is sent to the middle-tier object called ManagerServlet (coded using Java Servlet API) of *Intermediate Objects*. The ManagerServlet invokes another middle-tier object the search client (coded using LDAP API). The search is performed to trace the home of the User A in distributed *Data Repository* of 3-site testbed of *NPAS*. When the home of the User A is found, the *Site Profile Agent* at the local site (be it the home or visited site for the User A) is invoked by the SPAgentServlet (coded using Java Servlet API), another *Intermediate Object*. The SPAgentServlet creates a *Site Profile Agent*, which subsequently contacts the

corresponding (home or visited site of the User A) *Site Profile Agent*. The negotiation is carried out between *Site Profile Agents* to form a surrogate home for the User A at the visiting site. The guest profile is created (or updated, as the case may be) in the local *Data Repository* at both home and visited sites by the respective *Site Profile Agents*. All these activities take place without any interaction of the user; thus, it represents the server side of the implementation.

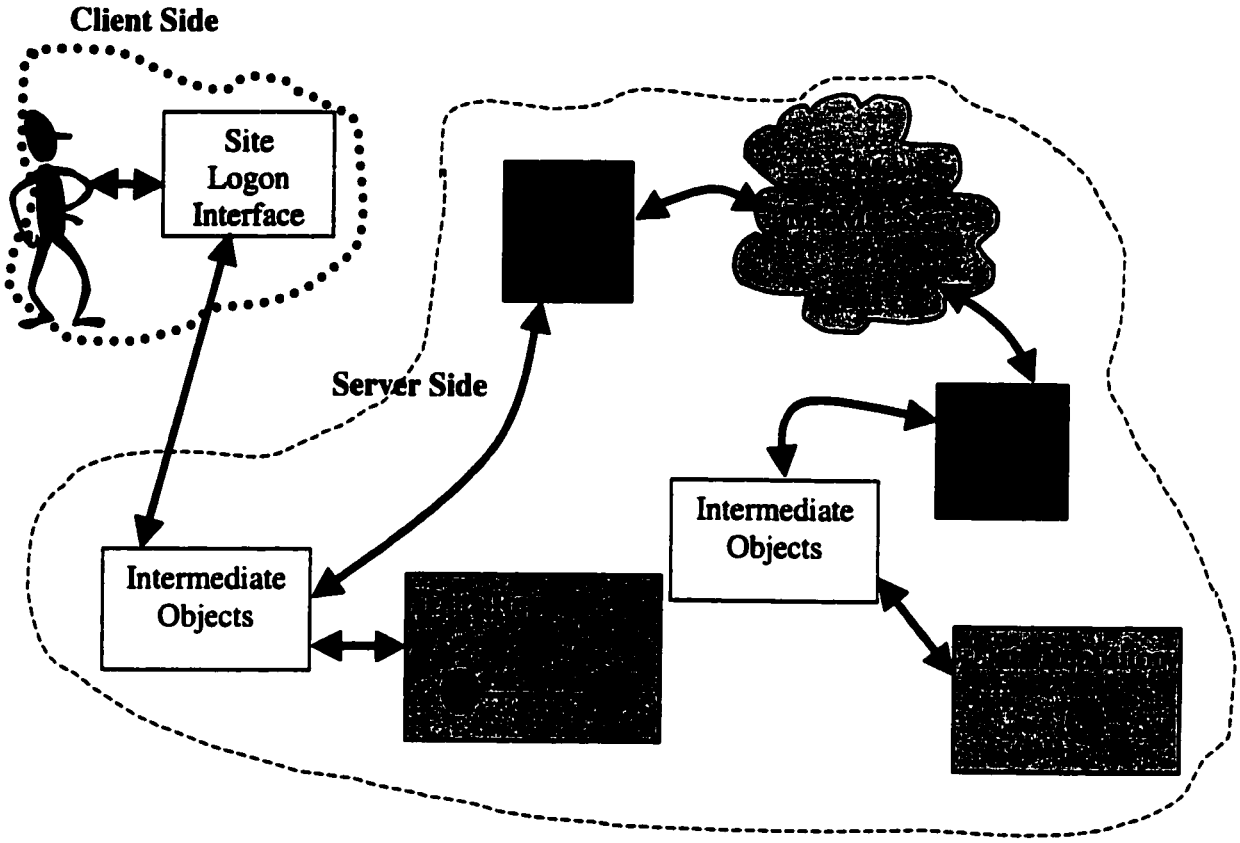


Figure 51: Implementation Overview

5.2 Data Repository

As discussed, during the analysis and design of *NPAS*, that *Data Repository* is one of the objects essential to the provision of *Messaging Services*. Further, for the reasons

described earlier, it was also noted the Internet's directory standard, i.e., LDAP is selected to implement the *Data Repository*. To set the context for the implementation details of the *Data Repository*, a brief overview of LDAP in relation to X.500 and its directory models precedes the presentation of *NPAS* data model.

5.2.1 LDAP in Relation to X.500 and its Models

As the name suggests, Lightweight Directory Access Protocol (LDAP) was conceived as a lightweight front end to X.500 servers; therefore, initially it only served at the client end, over TCP/IP connection. Nevertheless, LDAP client enjoyed full service of X.500 functionality and it spurred LDAP use as the Internet's directory service. However, the directory servers were still built on X.500 standard. Being the OSI directory service, X.500 implementation requires complex and heavyweight protocol stack, which makes it incompatible to the Internet. Thus, IETF then also provided stand-alone LDAP server that serves as a complete LDAP directory service, without requiring the use of X.500 at the back-end. To summarize, the IETF's LDAP standard differs from X.500 in the following respects [57]:

- LDAP runs directly over TCP, to avoid the overhead of the presentation and session layer of OSI stack.

- The LDAP has removed some less-used and redundant operations of X.500 (e.g., Read, List).

- Data elements of the LDAP directory are represented using string formats, simplifying the complex and highly structured representation of X.500.

- To transport data over network, LDAP uses simplified version of Basic Encoding Rules (BER); the same encoding scheme is used in X.500.

Although LDAP supports hierarchical organization of data (naming model), it is flexible enough to accommodate a single-level hierarchy. Moreover, the chaining mechanism of X.500 used for referrals (functional model of LDAP) stems a rigidly controlled infrastructure of higher-level (root, country, etc.) servers. The centralized registration of servers for creating global DIT contradicts the autonomy nurtured through the Internet. Thus, LDAP has adopted referral mode. According to which, the client is provided URL of another LDAP server when a server is unable to answer a query - the default TCP port for LDAP is 389. In LDAPv2, referral mode was not supported; however, some implementations (from Netscape and others) managed to put it in an error message field of the protocol. As of LDAPv3, referrals to other server or a set of servers may be returned [47]. According to Netscape Corp. the referral may be based on forward indexes.

With regards to security, X.500 provides robust mechanism to achieve authentication (knowing the party is, who they say they are) and privacy/access control (limiting information to the intended party). The X.500 standard supports simple authentication that is based on passwords and strong authentication that is based on public key cryptography. It also supports single entity authentication (a caller, i.e., client is authenticated to the called server) as well as mutual authentication (a caller and callee both authenticate to each other) [58]. However, LDAPv2 only supports simple authentication using cleartext password and Kerberos version 4. In LDAPv3, facilities are provided to support the simple authentication of LDAPv2 as well as Simple

Authentication and Security Layer (SASL). Further, LDAPv3 allows the server to return its credentials to the client (mutual authentication). The access control is not specified in LDAP standard. However, each implementation provides robust access control mechanisms, which may differ in format and capabilities [57].

The data/information model of LDAP is based on X.500 that defines what kind of information can be stored in the directory. Entries are basic building blocks of information model. An entry holds information about an object, which is of interest to the user of the directory. An object may be a real world thing such as information processing system (e.g., a mail server), telecommunication equipment or person. Directory object can be a mailing list containing names of people, organization profile, etc. An object class identifies similar objects, i.e., objects that have common characteristics. Every entry is a member of at least one object class. An entry consists of set of attributes. An attribute describes an aspect of that entry. The attribute type determines an attribute value. For example, an attribute of type 'telephone number' may take a value '613 562 5800'. The standard provides commonly used attributes. Additionally, user-defined attributes can also be created. Apart from the attributes typically used by users, directory contains attributes that are related to the operation and administration of directory. These are called operational attributes; e.g., the time the entry was last modified, access control attributes, etc.

Each entry in a directory has a distinguished name, which is formed by concatenating relative distinguished name of the entry with the name of its parent. A relative distinguish name uniquely identifies it from all its peers, which implies that distinguished name is unique in DIT.

5.2.2 NPAS Data Model

In the information model, as presented above, an object class is extensible by (single or multiple) inheritance. Analogous to object-oriented design, a class can be extended from the root object class or from an existing subclass. In either case, a subclass inherits all the mandatory and optional attributes of a parent class (es). The root object class in the X.500 standard is called top. To support the semantics of an application, a new object class can be created.

The *Data Repository*, as indicated in the previous chapter, requires three types of data: (i) *User-related*, (ii) *Workspace-related*, and (iii) *Site-related*. Before elaborating on them, it is appropriate to deliberate on some of the common attributes that are used in these categories.

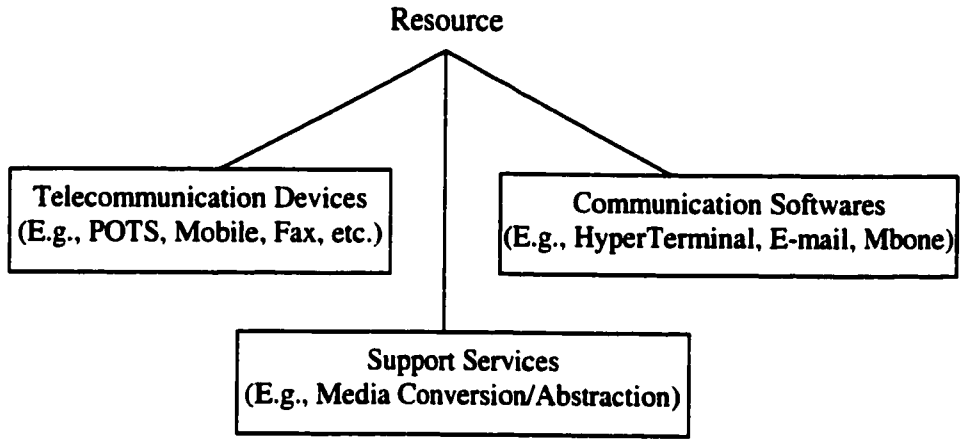


Figure 5.2: Types of Resources

5.2.2.1 Commonly-used Resources

A resource identifies a telecommunication device (e.g., phone, fax, pager), computing device (e.g., PDA, Laptop, Desktop) or information processing system (e.g.,

video abstraction server). Further, a resource comprises of a number of services, for example, a phone can be used to make an international long distance call, domestic long distance, call over PSTN connection, call over Internet, etc. Figure 5.2 shows the classification of resources that are accounted for in *NPAS* data model.

Although the current devices such as fixed telephone, pager, fax do not provide interface to authenticate user, rights and authentication attributes are specified for ideal devices. Further, it is noted that most of the pagers today only support one-way communication except for the state-of-the-art ones, which facilitate e-mail and text based two-way communication. Generally, communication devices are optionally provided with their individual inbox. However, in *NPAS*, as illustrated in the previous chapter, *Unified Message Box* replaces the multiple message inboxes that exist today.

1. Fixed Telephone

- Device Id: It contains country code, area code, local number, and where applicable extension number, as well. For example, 1 613 562 5800 x 6203.

- Service Level: Service levels may be (i) International long distance, (ii) Domestic long distance, (iii) PSTN Call, (iv) Internet Call. To further determine the specific details of a long distance plan, attributes may be (a) geographical areas, (b) time limits, (c) service provider name, etc. The voice mail associated to phone is redundant because all forms of communication under *Messaging Services* rely on *Unified Message Box* of *NPAS*, figure 4.9.

- Hardware Status: A device may be broken, working, or in-repair. Therefore, attributes would be: (I) Up, (ii) Down, (iii) In-Repair. An approximate 'downtime' (e.g., Date and Duration), should be stated when a device is in-repair.

- Rights: This identifies user privileges to receive or initiate calls. If a user has a right to initiate call, then service levels are also required. Moreover, to permit a user, authentication is also required. The privilege of the user will be discussed in *Site-related* data.

- Authentication: It captures attributes that allow authenticating a person who is permitted to use the device for receiving and/or initiating a call. The attributes could be unique user id and typical password or voice signature.

3. Fax Machine

- Device Id, Service Levels, Hardware Status, and Rights is defined in a similar manner as for a fixed telephone. However, authentication can be accomplished only for sending fax. In the case of a receipt of fax, it is assumed that *User Agent* does not send any high priority message to a fax machine, unless the user has asked to do so. Hence, authentication is not performed for receiving a fax.

2. Mobile Phone

- Device Id: It would be defined in a similar manner as fixed telephone id.

- Service Levels: Besides the above attributes, identified for fixed telephone service levels, it would also recognize if the phone is on roaming, i.e., has the user asked for mobility support beyond a local region or single operator.

- Hardware Status: It is defined in a similar manner as for fixed telephone.

- Rights: It is defined in a similar manner as for fixed telephone.

- Authentication: It is defined in a similar manner as for fixed telephone.

4. Pager

- Device Id and Hardware Status is defined in a similar manner as for a fixed telephone. However, two-way e-mail pagers that exist today have their proprietary addressing mechanism.

- Service Level: Numeric and alphanumeric service on one or two-way pagers are relevant to *NPAS*. However, voice-mail pagers, is not be used since *NPAS* provides its own *Unified Message Box*. The service levels defined for fixed telephone are applicable to pagers. Further, since a pager can also be activated for roaming, roaming is also an attribute.

- Rights: Depending on the type of pager (one or two-way) and on the privilege of the user, receive and initiate rights can be ascribed to a user.

- Authentication: It is defined in a similar manner as for fixed telephone.

5. Communication Software on Computing Devices

There are a variety of software programs available on Internet for text, voice and video communications. Most of them are built purely on client-server paradigm; servers for these software programs are only visible to users through proprietary clients programs. To limit the scope of data, the assumption here is that relevant servers are those that users mostly interact with to perform day-to-day communication at work, e.g., e-mail server.

In the context of client communication software, the options are: a computing device that has modem can connect to PSTN through HyperTerminal software; connecting from Internet to PSTN using a CTI-based application, among others. Also, an e-mail client is of interest to *NPAS*, and due to SMTP-based standardized implementation it is easy to know its attributes.

With regards to video-communication software, Mbone (Multicast Backbone Network) scales well for IP multicasts. Further, the concept of its session directory can be used to find online users. However, this functionality overlaps with the LMT of *NPAS*, where on the basis of the profiles for *Native* and *Visiting Mobile User* it is possible to discover devices/services that are in proximity of a mobile user. Mbone, like many other software on the Internet, identifies a user in its own way, introducing the additional mapping between unique id of *NPAS* and Mbone user id. To address it, a task manager (similar to the one on Windows NT) should register all communication software sessions running on a user station with the *Data Repository* of *NPAS*. Thus, *Data Repository* would contain the information on application(s) a user is working with. It is noted that Mbone diverges from the pure client-server model, using multicasts router that provides Mbone feed to the clients' software on end-systems.

A broad range of software programs exists (for interactive text chat, video conferencing) and each has created different identity for their users. Therefore, similar mechanism as suggested above in case of Mbone could be harnessed to integrate them into *NPAS Message Services*. However, it is not attempted here to capture their attributes.

5.1 Communication Sever Software on Workstations

- Service Id: The server id is the hostname/IP address, known port number, and protocol version number.

- Service Level: The service levels vary from software to software. In case of an e-mail server, the service levels are sending and receiving mails.

- Software Status: Also, need to monitor the health of server that provides messaging. Its status attributes could be the same as hardware status attributes.

- Hardware Status: It is defined in the same manner as for fixed telephone.

- Rights and Authentication: The rights specified for servers are: read and send messages. For authentication, it may require user account name and password for e-mail, for example.

5.2 Communication Client on Laptop/Desktop/PDA

- Device Id: The id in this case is a user id, e.g., e-mail address, chosen by the user or generated by a communication software application. The assumption is that the machine used by a mobile user already has installed the clients for those software applications.

- Service Level: In this case, QoS parameters are captured for the various applications, which are installed on a machine. For example, in case of e-mails, it needs to define parameters such as the size of mail that is permissible in a particular visited site/hosting network.

- Hardware Status: It is defined in the same manner as for fixed telephone.

- Rights and Authentication: The rights available for applications are: read and send messages. For authentication, it may require user account name and password for e-mail, for example.

It is a common practice nowadays to store applications configuration parameters and user's profile in one location, such as LDAP directory. This gives the benefit of maintaining a single copy of the data, allowing remote access.

6. Video Abstraction Service

This is an example of a special service that complements messaging applications. It is not possible to capture the attributes of this service until such a service exists. Nevertheless, on the basis of ongoing work in *MAA* [59], a few parameters are foreseen. It is assumed that such a service would be implemented as a server.

- Service Id: The server id is the hostname/IP address and known port number.

- Service Level: The service levels are: the video file formats supported by the service, expected processing time for varying accuracy and speed levels, etc.

- Status of Server: It is defined in the similar manner as for the fixed telephone.

- Rights: The rights to use this service means sending a video file for abstraction and receiving the abstract copy of it in return. Therefore, attributes are available or unavailable.

- Authentication: A human user or their *User Agent* may use this service; therefore, it requires user/agent id and password.

On the basis of the conceptual description of *NPAS* data model provided above, it is now possible to draw a set of attributes of commonly-used resources, while doing data reuse, where appropriate. All object classes and attributes definitions related to the *NPAS* data model conform to the "LDAPv3, Attribute Syntax Definitions"[60] and "A Summary of the X.500 (96) User Schema for use with LDAPv3" [61]. Further, the augmented Backus-Naur Form (BNF) notation which is used here to define the data model, is specified in "Standard for the Format of ARPA Internet Text Messages" [62].

- hardStatus: The value of the hardStatus attribute describes the status of the hardware of a device(e.g, telephone, host, etc). The hardStatus attribute is present in all objects of the subclass of device object class. The syntax values of this attribute are defined according to the following BNF.

```
Hard_Status        =     status-param
status-param       =     status-1_2 / status-3
status-1_2         =     "up" / "down"
status-3           =     ("in_repair" ["$"additional-param])
additional-param   =     (downSince "$" expectTobeUp)
downSince          =     Generalized Time; it is defined in RFC 2252,
                         section 6.14
expectTobeUp       =     Generalized Time
```

```
(NAME 'hardStatus' EQUALITY caseIgnoreMatch SUBSTR
caseIgnoreSubstringsMatch SYNTAX Hard_Status SINGLE-VALUE USAGE
userApplication)
```

- softStatus: The value of the softStatus attribute describes the status of the communication server software. The syntax values of this attribute are defined according to the following BNF.

```
Soft_Status        =     status-param
status-param       =     status-1_2 / status-3
status-1_2         =     "up" / "down"
status-3           =     ("in_repair" ["$"additional-param])
additional-param   =     (downSince "$" expectTobeUp)
downSince          =     Generalized Time; it is defined in RFC 2252,
                         section 6.14
expectTobeUp       =     Generalized Time
```

```
(NAME 'softStatus' EQUALITY caseIgnoreMatch SUBSTR
caseIgnoreSubstringsMatch SYNTAX Soft_Status SINGLE-VALUE USAGE
userApplication)
```

- POTSservicelevel: This attribute describes services: domestic, international long distance, PSTN/Internet connection, time limits, etc. in association to POTS. This attribute is used in fixedPhone object.

```
POTSserv Level     =     potservice-param
potservice-param   =     default / pot1 / pot2 / pot3
default            =     pot1
pot1               =     "regular_No_longdistance"
pot2               =     ("regular+longdistance" ["$" ("domestic" ["$"
                         (domesticlist)]) / ("overseas" "$"
                         overseaslist)]) ; if domestic is not followed
                         by optional parameters (domesticlist…), it
                         would mean long distance is countrywide.
pot3               =     "worldwide"
domesticlist       =     1#n(provinceOrState "$" timeLimits [ "$"
                         "pstn"/"internet"])
overseaslist       =     1#n(country "$" timeLimits ["$" "pstn" /
                         "internet"])
n                  =     dd
provinceOrState    =     st     ; st is defined in RFC 2256, section 5.9.
country            =     c      ; c is defined in RFC 2256, section 5.7
timeLimits         =     (dd ":"dd [":"dd] "$" dd ":"dd [":"dd])
```

```
(NAME 'POTSservicelevel' EQUALITY caseIgnoreMatch SUBSTR
caseIgnoreSubstringsMatch SYNTAX POTSserv Level USAGE userApplication)
```

- mobServicelevel: This attribute type extends POTSservicelevel attribute type, defined above. It includes roaming service as another aspect. This attribute is used in mobilePhone and pager objects.

```
Mobserv Level      =     mobservice-param
mobservice-param   =     ((default / mob1 / mob2 / mob 3) ["$" roam])
default            =     mob1
mob1               =     "regular_No_longdistance"
mob2               =     ("regular+longdistance" ["$" ("domestic" ["$"
                         (domesticlist)]) / ("overseas" "$"
                         overseaslist)]) ; if domestic is not followed
                         by optional parameters (domesticlist…), it
                         would mean long distance is countrywide.
domesticlist       =     1#n(provinceOrState "$" timeLimits [ "$"
                         "pstn"/"internet"])
overseaslist       =     1#n(country "$" timeLimits ["$" "pstn" /
                         "internet"])
n                  =     dd
provinceOrState    =     st     ; st is defined in RFC 2256, section 5.9.
country            =     c      ; c is defined in RFC 2256, section 5.7
timeLimits         =     (dd ":"dd [":"dd] "$" dd ":"dd [":"dd])
roam               =     [domestic][overseas] (domesticlist /
                         overseaslist ) / (domesticlist "$"overseaslist)

(NAME 'mobServicelevel' EQUALITY caseIgnoreMatch SUBSTR
caseIgnoreSubstringsMatch SYNTAX Mobserv Level USAGE userApplication)
```

- phoneFixMobFaxPager: This attribute defines: is the device a fixed phone, mobile phone, fax machine or a pager. This attribute is present in all the objects of telecommunication device type. Its syntax value is defined according to the following BNF.

```
Device?            =     "fixedPhone" / "mobilePhone" / "faxPhone" /
                         "pager"

(NAME 'phoneFixMobFaxPager' EQUALITY caseIgnoreMatch SYNTAX Device?
SINGLE-VALUE USAGE userApplication)
```

- sharedUse: This attribute describes: should the device be used for strictly personal, shared or occasionally shared use. This attribute is present in all devices, e.g., phone, pager. It syntax values is defined according to the following BNF.

```
Use MobOrFix       =     usemof-param
usemof-param       =     ("strictly-personal"["$" distinguishName])
                         /("occasionally- shared" ["$" distinguishName])
                         / "collective"
distinguishName    =     DN     ; It is distinguished name of Native
                         Mobile User, the syntax is defined in RFC 2252,
                         section 6.9.

(NAME 'sharedUse ' EQUALITY caseIgnoreMatch SYNTAX Use MobOrFix SINGLE-
VALUE USAGE userApplication)
```

- mediaCapabilities: This attribute explicitly describes the hardware capability of a device to present different media, e.g., text, voice, image, and video. This attribute is present in all the subclasses of device object class.

```
Media Capab        =     media-param
media-param        =     ["voice"] ("$"[displaysize "$" typeOfdisplay
                         "$" resolution])
displaysize        =     (d[d] "*" d[d] ("inch" / "cm"))
typeOfdisplay      =     printablestring
resolution         =     d[dddd] "*" d[dddd]

(NAME 'mediaCapabilities' EQUALITY caseIgnoreMatch SUBSTR
caseIgnoreSubstringsMatch SYNTAX Media Capab SINGLE-VALUE USAGE
userApplication)
```

- emailServicelevel: This attribute type classifies the file sizes for the transfer of large multimedia attachments and it also contains priority levels to be used for mail-delivery. Its syntax is defined according to the following BNF.

```
Email Servlevel    =     email-param
email-param        =     *1(["small" / "large" / "any"]) ["$"
                         *1(priority)]     ; where 'small', 'large' and
                         'any' signifies the size of files
priority           =     "low" / "normal" / "high"     ; the priority of
                         levels for mail delivery

(NAME 'emailServicelevel' EQUALITY caseIgnoreMatch SYNTAX Email
Servlevel USAGE userApplication)
```

- serverId: This attribute defines the host name and port number of a communication server software. Its syntax value is defined according to the following BNF.

```
Serv ID            =     (hostname / IPAddress "$" portNumber))
hostname           =     printablestring
IPAddress          =     d[dd] "." d[dd] "."d[dd] "." d[dd]
portNumber         =     dddd[d]

(NAME 'serverId' EQUALITY caseIgnoreMatch SYNTAX Serv ID SINGLE-VALUE
USAGE userApplication)
```

- currentUserActivity: This attribute describes what communication software(s) a user is running at a current time while on a computing device. This data is collected by a task manager program at a user end system and is stored in a user's profile (*Native*

Mobile User or *Visiting Mobile User*). The syntax value of this attribute is defined according to the following BNF.

```
Current Sessions  =     session-param
session-param     =     (1#software "$" sessionid)
software          =     "E-mail" / "Mbone" / "Chat Program"
sessionid         =     printablestring

(NAME 'currentUserActivity' EQUALITY caseIgnoreMatch SYNTAX Current
Sessions USAGE userApplication)
```

- videoServiceLevel: This attribute describes the service levels supported by a video abstraction server software. Its syntax value is defined according to the following BNF.

```
Video Level       =     serv-param
serv-param        =     (fileFormats ["$" AccuracyLevels])
fileFormats       =     1#format
accuracyLevels    =     "normal" / "advanced"
format            =     "AVI" / "MPEG" / "QTW"

(NAME 'videoServiceLevel' EQUALITY caseIgnoreMatch SYNTAX Video Level
USAGE userApplication)
```

- interaction: This attribute type defines the communication activities that may be performed using a communication device, either when a user is local to or remote from their home site.

```
Interact          =     ((1#choice) ["$" default])
choice            =     ("send") /("receive")    ; when right is "send",
                        then service levels must be specified
default           =     ("storeInmailbox" "$" "retrieveFrommailbox")

(NAME 'interaction' EQUALITY caseIgnoreMatch SYNTAX Interact USAGE
userApplication)
```

- authenticationMethod: The value of authenticationMethod attribute describes the level of authentication to be used before allowing the use of a hardware or software resource. This attribute is present in all the subclasses of device object class. The syntax values of this attribute are defined according to following BNF.

```
Auth Method       =     auth-param
auth-param        =     auth1 /auth2 / auth3
```

```
auth1              =      "none"
auth2              =      "weak"
auth3              =      ("strong"[ "$" enryptionScheme])
encryptionScheme   =      algorithm parameters

(NAME 'authenticationMethod' EQUALITY caseIgnoreMatch SUBSTR
caseIgnoreSubstringsMatch SYNTAX Auth Method USAGE userApplication)
```

It should be noted that security of *NPAS* in not in the scope of this thesis work. Thus, it is not attempted here to specify fully the values of this attribute: auth-param may have more than three levels of security, and algorithm parameters are left unspecified.

- npasDevice: It is a parent object class for telecommunications, computing devices that run communication software and support services. However, it is subclass of device object class, which is defined in RFC 2256, based on X.521. The device object class is used to define entities, e.g., modem, telephone and disk drive. The npasDevice object class contains the attributes that will be inherited by its child object classes.

The schema definition of the npasDevice object class:

Name of Object Class	npasDevice	
Definition	It's a parent object class for Telecommunication Devices, Communication Software and Support Services	
SUBCLASS OF	device	It is defined in RFC 2256, section 7.15
MUST CONTAIN	hardwareStatus	It is defined above.
MAY CONTAIN	authenticationMethod	It is defined above.
	mediaCapabilities	It is defined above.
	sharedUse	It is defined above.

- npasTeleDevice: It represents a telecommunication device, namely fixed telephone, mobile phone, fax machine, and a pager. Therefore, it is a subclass of npasDevice. The attributes are drawn from the dimensions DeviceId.

The schema definition of npasTeleDevice object class:

Name of Object Class	npasTeleDevice	
Definition	It represents a telecommunication device.	
SUBCLASS OF	npasDevice	It is defined above.
MUST CONTAIN	telephoneNumber	This attribute is defined in RFC 2256.
MAY CONTAIN	extensionNumber	It is a new attribute type defined for NPAS. Its value can be a numeric string.
	phoneFixMobFaxPager	It is defined above.

- npasServer: It represents a communication software server, which is considered as a device. Therefore, it is a subclass of npasDevice. The attributes are drawn from the dimension ServiceId. Several other attributes, which are already known from Netscape's proprietary schema, for example, serverProductName, serverVersionNumber, installationTimeStamp, administratorContactInfo, adminURL, are not shown here.

Since, email server is a type of communication software server; therefore, it can be represented as a npasServer.

The schema definition of npasServer object class:

Name of Object Class	npasServer	
Definition	It represents a software server	
SUBCLASS OF	npasDevice	It is defined above.
MUST CONTAIN	serverId	It is defined above.
	softStatus	It is defined above.

- npasVideoAbs: This object class represents the video abstraction server. Semantically it is not a type of communication software server; however, the base class of npasServer is generic enough to be considered just as a server. And, since

npasVideoAbs is a server, therefore, it is defined as a subclass of npasServer. The attributes are drawn from the dimension ServiceLevel.

The schema definition of npasVideoAbs object class:

Name of Object Class	npasVideoAbs	
Definition	It represents a video abstraction server.	
SUBCLASS OF	npasServer	It is defined above.
MAY CONTAIN	videoServicelevel	It is defined above.

5.2.2.2 Site-related Data

Although, *User-* and *Workspace-related* data has appeared prior to *Site-related* data when first introduced in section 4.4.2, here the order has been completely reversed. Since, *Site-* and *Workspace-related* data provide the basis to define the attributes described in *User-related* data; their description precedes it.

Site-related data contains organization's policies that safeguard the security of communication resources owned by an organization. These policies/regulations framework should:

i. facilitate the accountability of proper usage of communication resources,

ii. maximize the availability of communication access for users, for example those who have high-precedence over others should not be held back by low-priority users,

iii. support the nomadic moves of their employees/affiliated users to different organizations,

iv. provide security definitions for accessing a device and its corresponding services,

v. facilitate the creation of varied secure regions within organization: departments or units, as an example.

As it is evident, providing a framework for defining organization policies is in itself a significant task. Therefore, here attempt is only made to capture some dimensions of *Site-related* data, and on the basis of those a few attributes and object classes are drawn:

- Services: It describes services and their load limits.
- Restricted Areas: It describes secure zones in the organizational workspaces that can only be assigned to privileged users.
- Authentication Levels: It describes the different authentication types to be used as security is heightened.
- Service Contracts: It describes the kinds of agreement for both native user (user who is an employee of the organization but may become a visitor at other organization) and visiting user (user who is visiting from the other organization).

It is *Site-related* data that specifies what are protected resources and what authentication levels are required before the service access rights will be granted, which is similar to access control defined for X.500. The *Site-related* data differs from access control, since it not only provides access control semantics but also, most importantly, determines what access control is applicable in different situations. In other words, when *Site-related* data is put into the system, the profiles that are assigned to users could be approved, disapproved, or alternate suggestions can be made. However, the LDAP

directory has no mechanism to express semantic approval or disapproval, based on the constraints charted by some data that is already in the directory, on creating new data that is to be put into the directory. Thus, the aim here is to create *Site-related* data that is consulted by agents, for example, to create the profiles for visitors, to reserve resources before relaying multimedia messages, etc., the idea is illustrated in Figure 5.3. It is assumed here that policy data is local to each site; however, there should be policy data that is understood across organizations. Although, it is not addressed here, except in a weak sense.

In the context of *Site Profile Agents* negotiation, it is of immediate interest to have the policy data that comprises of classification of native and visiting users and services hosted by an organizational site. Therefore, the rest of discussion would elaborate on it.

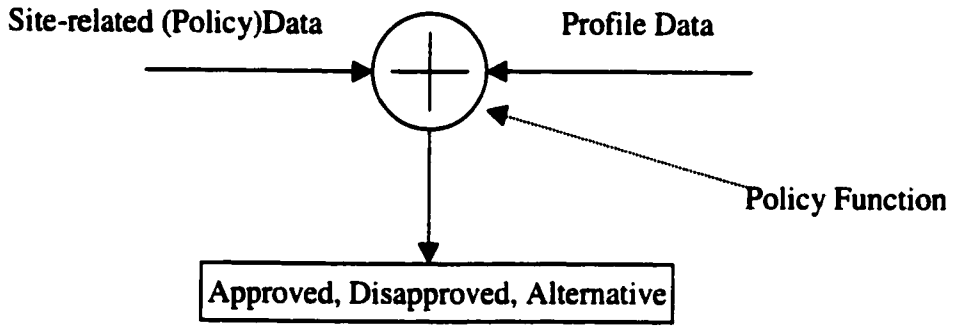


Figure 5.3: Role of Policy Data

It is assumed that the services that an organization provides are either offered directly by it or purchased from other organizations. The purchased services would be consumed either when the home organization does not offer those services or a

native/visiting user is farther from the local networks of their home organization and it is cheaper to use purchased services.

- npasResrceDef: This object class defines the kinds of resources an organization (workspace in an organization) offers, directly or externally.

The schema definition of npasResrcDef object class:

Name of Object Class	npasResrcDef	
Definition	It represents a resource definition.	
SUBCLASS OF	top	
MUST CONTAIN	orgOrWorkspace	It is defined below.
MAY CONTAIN	resrcDefintion	It is defined below.

- orgOrWorkspace: This attribute type gives the distinguished name of the space that hosts the resources listed in the resrcDefinition attribute type. It syntax value is defined according to following BNF:

```
Org-Workspace      =     DN    ; it is distinguished name of the
                         organization or workspace, the syntax is
                         defined in RFC 2252, section 6.9.

(NAME 'orgOrWorkspace' EQUALITY caseIgnoreMatch SYNTAX DN USAGE SINGLE-
VALUE userApplication)
```

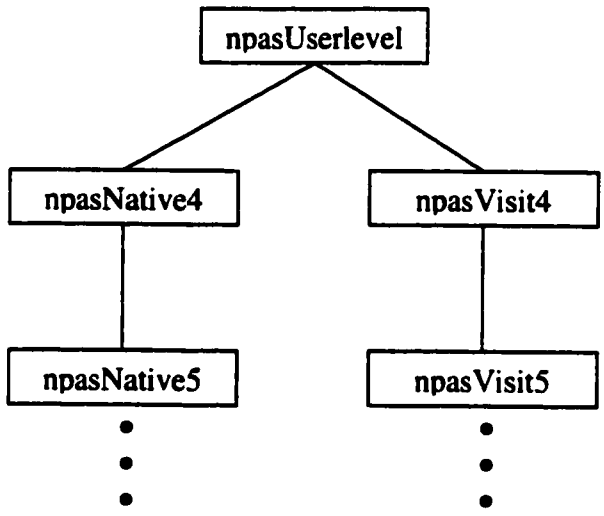
- resrcDefinition: This attribute type describes the kind of resources that may be offered by an organization, directly or externally.

```
Resource Definition=    ["direct"/"purchased"] 1#(res ["$" interaction
                        "$" serv-level ])       ; certain decision
                        requires the knowledge of whether a service is
                        direct or purchased. When visitor requests for
                        a purchased service, it may not be offered, for
                        example.
interaction        =    The interaction attribute type is already
                        defined above.
res                =    "fixedPhone" / "mobilePhone" / "email" /
                        "videoAbs" / "videoConf" / "faxPhone" / "pager"
serv-level         =    POTSservicelevel / mobServicelevel /
                        videoServicelevel / emailServicelevel     ;
                        these attributes types are already defined
                        above.
```

```
(NAME 'interaction' EQUALITY caseIgnoreMatch SYNTAX Resource Definition
USAGE userApplication)
```

- permission: This attribute type defines whether a user is either allowed or disallowed to use the resources described by resrcDefinition attribute type. The syntax value of this attribute is defined according to following BNF.

```
Permits          =      "allowed" / "disallowed"
```




```
(NAME 'permission' EQUALITY caseIgnoreMatch SYNTAX Permits USAGE
userApplication)
```

Figure 5.4: npasUserlevel Object Hierarchy

- npasUserlevel: This is a concrete object class that defines the classification for native and visiting users. It is assumed that a native or visiting user is an organizational person (i.e., affiliated to an organization) rather than a free-lance visitor. A privilege reflects that a user is granted or forbidden access to a service. The types of privileged users can be defined on the scale of 4 to 6, for example. The number 4 identifies the least privileged user and the number 6 identifies the most privileged user. To elaborate, a native user is may be associated to one of the objects i.e., npasNativeU4, npasNativeU5, or npasNativeU6. Similarly, a visiting user is either a member of

npasVisitU4, npasVisitU5, or npasVisitU6. Figure 5.4 illustrates the hierarchy of objects of npasUserlevel class.

The schema definition of npasUserlevel object class.

Name of Object Class	npasUserlevel	
Definition	It represents a parent class of user privilege classification.	
SUBCLASS OF	top	It is defined in RFC 2256, section 7.1.
MUST CONTAIN	orgOrUnit	It is defined below.
	levelName	It is defined below.
MAY CONTAIN	aPrivilege	It is defined below.

- orgOrUnit: This attribute type contains the name of organization or its organization unit. Its value is defined in the following BNF.

```
OrgOrUnt          =     o / ou              ; o and ou is defined in RFC
                        2256, section 5.11 and 5.12, respectively.

(NAME 'orgOrUnit' SUP name)
```

- levelName: This attribute type contains name of the instance of the npasUserlevel e.g., npasNative4, npasVisit4, etc.

```
Level Name        =     npasNative4/npasNative5/npasNative6/npasVisit4/
                        npasVisit5/npasVisit6

(NAME 'levelName' EQUALITY caseIgnoreMatch SYNTAX Level Name USAGE
SINGLE-VALUE userApplication)
```

- aPrivilege: This attribute describes which resrcDefinition is allowed or disallowed. The npasUserlevel object class would contain multiple copies of this attribute. Its syntax value is defined according to following BNF.

```
Privilege         =     permission "$" rsrcDefintion  ; rsrcDefinition
                        and permission attribute types are described
                        above.

(NAME 'privilege' EQUALITY caseIgnoreMatch SYNTAX Privilege USAGE
userApplication)
```

- npasNative4: This object is an instance of npasUserlevel; it illustrates the minimum set of privileges that may be assigned to a native user of ou=SITE, o=UOttawa, c=CA. The set of privileges assigned to npasNative4 is:

```
aPrivilege  =   allowed$direct$fixedPhone$send,receive$storeInmailbox
                $retrieveFrommailbox$regular_no_longdistance
aPrivilege  =   allowed$direct$faxPhone$send,receive$storeInmailbox$
                retrieveFrommailbox$worldwide
aPrivilege  =   allowed$direct$emai$send,receive$worldwide
aPrivilege  =   allowed$direct$videoAbs$AVI,MPEG,QTW$normal
```

- npasNative5: This object is instance of npasUserlevel; it illustrates the next set of privileges that may be assigned to a native user of ou=SITE, o=UOttawa, c=CA.

The set of privileges assigned to npasNative5 is:

```
aPrivilege  =   allowed$direct$fixedPhone$send,receive$storeInmailbox
                $retrieveFrommailbox$regular+longdistance$domestic
aPrivilege  =   allowed$purchased$mobilePhone$send,receive$
                storeInmailbox$retrieveFrommailbox$regular+longdistan
                ce$domestic$domestic
aPrivilege  =   allowed$direct$faxPhone$send,receive$storeInmailbox$
                retrieveFrommailbox$worldwide
aPrivilege  =   allowed$direct$email$send,receive$worldwide
aPrivilege  =   allowed$direct$videoAbs$AVI,MPEG,QTW$normal
```

- npasNative6: This object illustrates the set of maximum privileges that may be assigned to a native user of ou=SITE, o=UOttawa, c=CA.

The set of privileges assigned to npasNative6 is:

```
aPrivilege  =   allowed$direct$fixedPhone$send,receive$storeInmailbox
                $retrieveFrommailbox$worldwide
aPrivilege  =   allowed$purchased$mobilePhone$send,receive$
                storeInmailbox$retrieveFrommailbox$worldwide$
                worldwide
aPrivilege  =   allowed$direct$faxPhone$send,receive$storeInmailbox$
                retrieveFrommailbox$worldwide
aPrivilege  =   allowed$direct$email$send,receive$worldwide
aPrivilege  =   allowed$direct$videoAbs$AVI,MPEG,QTW$normal
```

It can be noted that the data representation scheme and attributes captured thus far provide a flexible structure to generate a variety of privileges. Similar to native users,

npasVisit4, 5, and 6 can be formed. The assignment of privileges at this point has been done arbitrarily to illustrate the use of data model. Semantically, however, its values are derived from the management polices of an organization. Therefore, it can be suggested that there are a number of policy levels. That is, the npasUserlevel acts as a policy data that approves the privilege assignment for a native or visiting user. Further, the objects of type npasUserlevel itself can be generated from a higher-level policy data, automating the creation of arbitrary instances of npasUserlevel. These threads of *Site-related* data needs further investigation, which is not addressed in this thesis work.

5.2.2.3 Workspace-related Data

This category of *NPAS* data model provides information pertaining to physical resources available in a location/workspace area within an organization. A workspace either corresponds to a real world workspace (office) or a logically defined area in a large space. A variety of communication and computing devices could be available in a workspace. The object classes of *Workspace-related* data category are: npasDevice, npasTeleDevice, npasServer, npasEmailServer, npasVideoAbs, which have been described above. There is another object class in this category which is called npasWorkspaceOrEquivalent. Its schema provides attributes to store addresses of telecommunication and computing devices associated with a workspace area. The data in workspace category is statically populated and is long-lived. The data in workspace-related data category enables *Site Profile Agent* to identify the shared and/or purchased resource(s) that a *vmUser* could use.

The schema definition of npasWorkspaceorEquivalent object class:

Name of Object Class	npasWorkspaceorEquivalent	

Definition	It represents a parent class of user privilege classification.	
SUBCLASS OF	top	It is defined in RFC 2256, section 7.1.
MUST CONTAIN	locationId	The room where people work. It is defined in RFC 1274. Its value follows SYNTAX caseIgnoreMatch. It will be locally defined how to write the room number i.e. building abbreviation, floor #, section of the floor, and the room #, e.g., IIT 286 or M-50 286.
	cn	It is defined RFC 2256, section 5.4.
	owner	It is defined in RFC 2256, section 5.33.
	ownersMail	
MAY CONTAIN	description	It is defined in RFC 2256, section 5.14.
	npasTeleDeviceDN	This identifies the distinguished name of npasTeleDevice in a workspace. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	npasEmailServerDN	This identifies the distinguished name of npasEmailServer in a workspace. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	npasVideoAbsDN	This identifies the distinguished name of npasVideoAbs in a workspace. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	resrcLoadLimit	It is defined below.

- resrcLoadLimit: This attribute type defines load limitation related to the use of the shared use device available in a workspace. This attribute is semantically associated to subclasses of npasDevice.

```
Load Limit        =     (dd "$" resrcDefintion "$" DN)       ; where dd
                        is the number of users that can be mapped to
                        the same shared device and DN is the
                        distinguished name of the device
```

```
(NAME 'resrcLoadLimit' EQUALITY caseIgnoreMatch SYNTAX Load Limit USAGE
userApplication)
```

5.2.2.4 User-related Data

User-related data comprises of identity, native and guest privileges of a nomad, etc. It provides attributes to store the dynamic data on changing locations and the corresponding devices available to a nomad in that location. Therefore, data in this category is of static and dynamic nature. Not all of the *User-related* data can be stored in the *Data Repository*; for example, the data pertaining to the message forwarding/screening (which would be rules/filters) should be stored in the knowledge base of *User Agent*. The resources required to receive and send messages (the function of *Messaging Services*), referred to as 'resource-related preferences', should be stored in the *Data Repository*. Two object classes are created to encompass the two roles of a mobile user: *Native Mobile User* (*nmUser*) and *Visiting Mobile User* (*vmUser*).

Native Mobile User (nmUser): A *Native Mobile User* refers to a mobile user who has subscribed to their home site, which is an *Enabled* site. The npasNMUser object class defines schema to store attributes describing user's personal and shared resources at their home location, their npasUserlevel, resource-related preferences, etc. It also includes the learned data on user's location, preferences and messaging requirements at home and visited sites; however, these data is stored in the knowledge base of the *User Agent*. The *User Agent* [36] can also supply the dynamic values for attributes declared in npasNMUser to build a personalized service profile for the user. This class is also suitable to store nomad's itinerary history for predictive mobility, not detailed here though. This class offers data to *Messaging Services* agents to determine an alternative resource. For example: should the nomad's mobile phone be dead, it allows finding a

desk phone in proximity. The attributes defining the object class npasNMUser fall into three categories: (i) user privileges; (ii) resources at home and visited sites; and (iii) resource-related preferences.

Resources at home and visited sites: These are (addresses of) resources, provided from a home/visited site, that a user is permitted to use. The resources may be fixed telephone, mobile telephone, pager (one or two-way), fax, laptop, desktop, PDA, video abstraction service.

Resource-related preferences: These represent the types of required and desired devices that is determined by a user on the basis of the nature of expected incoming messages, the length of stay at a visited site, among others. The attributes describing resource-related preferences are:

- reqDevicelist: This attribute contains the list of required devices/resources requested by a *Visiting Mobile User* when at a visited site. Its syntax value is defined according to the following BNF.

```
ReqDevices          =     req-param
req-param           =     1#dev
dev                 =     "fixedPhone" / "mobilePhone" / "pager" / "fax"
                          / "email" / "personalComputer"

(NAME 'reqDevicelist' EQUALITY caseIgnoreMatch SYNTAX ReqDevices USAGE
userApplication)
```

- desDevicelist: This attribute contains the list of desired devices/resources requested by a *Visiting Mobile User* when at a visited site. Its syntax value is defined according to the following BNF.

```
DesDevices          =     des-param
des-param           =     1#dev
dev                 =     "fixedPhone" / "mobilePhone" / "pager" / "fax"
                          / "email" /"personalComputer"
```

```
(NAME 'desDevicelist' EQUALITY caseIgnoreMatch SYNTAX DesDevices USAGE
userApplication)
```

The schema definition of npasNMUser object class:

Name of Object Class	npasNMUser	
Definition	It represents a Native Mobile User.	
SUBCLASS OF	organizationalPerson	This object class is defined in RFC 2256, section 7.8.
MUST CONTAIN	homeMail	This is email address of the user., which is unique globally. Its SYNTAX is MHS or Address, defined in RFC 2252, section 6.20. EQUALITY is caseIgnoreMatch.
	locationId	It is defined above under npasWorkspaceorEquivalent objet class.
	npasUserlevelDN	This identifies the distinguished name of npasNative4, 5 or 6, defined above. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
MAY CONTAIN	homeTelephoneDN	This identifies the distinguished name of fixedPhone in a workspace. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	mobileDN	This identifies the distinguished name of mobilePhone in a workspace. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	pagerDN	This identifies the distinguished name of pager in a workspace. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	homefaxDN	This identifies the distinguished name of faxPhone in a workspace. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	reqDevicelist	It is defined above.
	desDevicelist	It is defined above.
	aVisitprivilege	This identifies the requested services associated to desired and/or required device list, when visiting a site. Its syntax is same as aPrivilege, defined above.
	visitTelephoneDN	This identifies the distinguished name of fixedPhone at a visited site. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	visitmobileDN	This identifies the distinguished name of mobilePhone at a visited site. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	visitpagerDN	This identifies the distinguished name of pager at a visited site. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	visitfaxDN	This identifies the distinguished name of faxPhone at a visited site. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	visitSiteServiceGateway	This attribute type describes a Site Service Gateway (section 4.5.1). It syntax value is

		defined according to BNF of server ID.
	currentUserActivity	It is defined above.

Visiting Mobile User (vmUser): A *Visiting Mobile User* parallels *nmUser* at a visited location. Therefore, its object class stores similar attributes as to npasNMUser. The class holds data on the resources that are approved for the use of a *vmUser* at a visited location. The data in npasVMUser is presumed to have a shorter lifetime as compared to npasNMUser. Therefore, there is an attribute type in this object class called accDeletime (i.e.. account deletion time), that specifies a time limit for the deletion of privileges and data from the visited site. The syntax of this time is defined according to following BNF.

```
Time Limit          =      time-param ":" time-param
time-param          =      Generalized Time

(NAME 'accDeletime' EQUALITY caseIgnoreMatch SYNTAX Time Limit USAGE
userApplication)
```

The schema definition of npasVMUser object class:

Name of Object Class	npasVMUser	
Definition	It represents a Visiting Mobile User.	
SUBCLASS OF	organizationalPerson	This object class is defined in RFC 2256, section 7.8.
MUST CONTAIN	homeMail	This is email address of the user., which is unique globally. Its SYNTAX is MHS or Address, defined in RFC 2252, section 6.20. EQUALITY is caseIgnoreMatch.
	locationId	It is defined above under npasWorkspaceorEquivalent objet class.
	npasUserlevelDN	This identifies the distinguished name of npasVisit4, 5 or 6, defined above. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
MAY CONTAIN	homeTelephoneDN	This identifies the distinguished name of fixedPhone in a workspace. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	mobileDN	This identifies the distinguished name of mobilePhone in a workspace. Its SYNTAX is DN, defined in RFC 2252,

		section 6.9.
	pagerDN	This identifies the distinguished name of pager in a workspace. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	homefaxDN	This identifies the distinguished name of faxPhone in a workspace. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	visitTelephoneDN	This identifies the distinguished name of fixedPhone at a visited site. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	visitmobileDN	This identifies the distinguished name of mobilePhone at a visited site. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	visitpagerDN	This identifies the distinguished name of pager at a visited site. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	visitfaxDN	This identifies the distinguished name of faxPhone at a visited site. Its SYNTAX is DN, defined in RFC 2252, section 6.9.
	visitSiteServiceGateway	This attribute type describes a Site Service Gateway (section 4.5.1). It syntax value is defined according to BNF of server ID.
	currentUserActivity	It is defined above.

5.2.3 Deployment of NPAS Data Model

The data model presented above is an extension of standard directory schema and conforms to the extension mechanisms provided in LDAP. In this work, Netscape's Directory Server has been used to deploy the *Data Repository* for *NPAS's* 3-site testbed. Given the availability of evaluation version and early provision of Java LDAP API, Netscape's LDAP server implementation was selected to implement *Data Repository*. The implementation process includes the update of user-schema configuration files for object classes and attributes, a file that defines the nodes of DIT and associated object classes and the search filter configuration files. There are files that need to be created in order to extend the web-gateway provided by Netscape through its Enterprise Server. This requires creation of html-like files. According to Netscape, no provisions are made to support presentation of web interface for extended schema. Therefore, the html-like

format (server-side html) of built-in files that comes with Directory Server schema was used as a template to write new html-like files.

5.2.4 Qualitative Evaluation of X.500-Based Data Model

It can be noted that object-oriented data model of X.500 provides extensibility: new features can be created by specialization of existing features. Since, X.500-based data model supports a hierarchical naming space (DIT) it is easy to add more organizations into testbed. This implies that the *NPAS* environment would require a hierarchically distributed model of LDAP to achieve a wider coverage. This results in modifying the flat distributed directory model, which is being used currently. This assumption has its roots in the existing LMT*s* for terminal mobility [24],[26]. In addition, with the hierarchical *Data Repository* model the outgoing messages from the enabled site do not get sent directly to the recipient's home until search operation traverses up in the directory levels, which eliminates routing redundancy.

A directory is not intended for frequent updates, rather it is optimized for search queries. Therefore, dynamic data in *NPAS* data model puts performance demands on back-end database of LDAP implementations. Nonetheless, the LDAP (v3) was proposed to support dynamic entries as suggested in the Internet-Draft of LDAP v3; however, it did not sift through the Standard Track of LDAP v3.

5.3 Site Logon Interface

According to the design objectives of the *Site Logon Interface* set earlier, it is implemented as a Java applet. A Java applet affords (i) ubiquitous network access, (ii) thin-client requirement, and (iii) secure and friendly interface. However, the primary

emphasis of the implementation in this attempt is to determine the minimal input required of a user for their location registration. The figure 5.5 shows a snapshot of the interface.

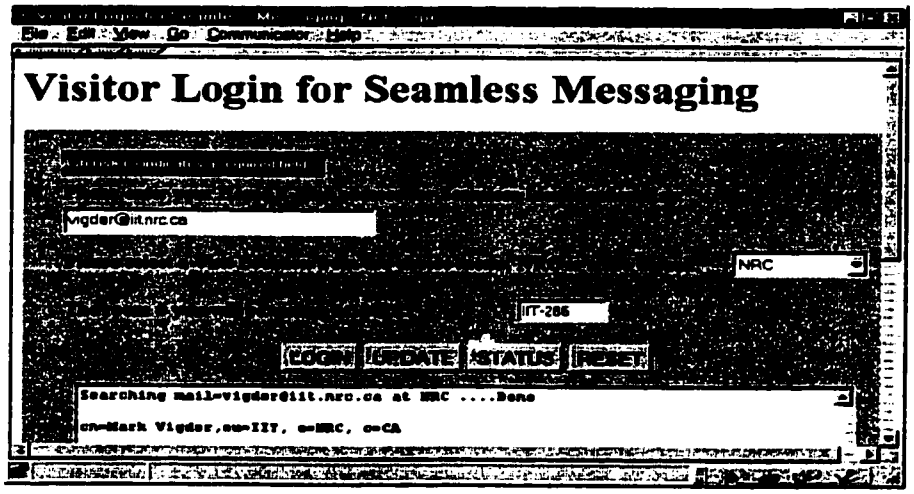


Figure 5.5: A Snapshot of Site Logon Interface

A user is required to input his or her home e-mail address, which is selected to be the unique identity of every *NPAS* subscriber. Then, the user selects the site he or she intends to visit, followed by the location ID of a room. Subsequently, the user presses LOGIN button to send this data for location registration. The result of the login appears in the text area of the applet (based on the LMT described earlier). The UPDATE button is to be used when a user wants to update location ID within the visited site after the first logon. The STATUS button allows checking the current profile of the user at any time after logon. Finally, the RESET button sets the user input fields to dummy values. The UML class diagram for applet code and related interaction diagrams are shown in section 5.5.

5.4 Site Profile Agent Negotiation

The *NPAS* data model, as stated earlier, not only forms the basis for *Messaging Services*, but also *Site Profile Agents'* negotiation relies on the data stored in the *Data Repository* to create *vmUser's* guest profile. It should be also noted that the *Site Profile Agent* serves the role of the policy function, shown in figure 5.3, when creating a *vmUser's* profile in respect of the policies and authorizations of the visited site. Based on the design objectives, the high-level negotiation strategy described in the previous chapter and the *NPAS* data model presented above, this section provides the details of the negotiation.

The *Site Profile Agents'* negotiation is defined off-line [34] as a pair of caller and callee conversations. The agent that initiates a conversation is identified as caller and the one that responds to it is called callee. A conversation can be defined, for example, using Definite Clause Grammar (DCG) [23] or state transition diagram [63], [34]. The conversation is defined here using finite state machines.

Figure 5.6, shows home-foreign *Site Profile Agent* negotiation through Mitel Corporation's blackboard, which is known as Micmac [64]. Assume that Site 1 and Site 2 are enabled sites and the user 'A' is the *Native Mobile User* of Site 1(i.e., Site 1 is her home). The user 'A' has logically two logon options: she can logon to Site 2 using home site logon interface irrespective of whether she is physically at home or not. As a second option, she can logon through visited site logon interface irrespective of whether she is physically present at visited site or not. In the first case, she logs on to Site 2 through her home site logon interface. Thereby, her home *Site Profile Agent*, as a caller, initiates negotiation addressing Site 2's *Site Profile Agent*, it is called here a foreign *Site Profile Agent*, acting as the callee for this scenario. In the second case, she may log on to Site 2

through the visited site logon interface. Site 2's *Site Profile Agent*, the foreign *Site Profile Agent*, acting as a caller, initiates negotiation addressing her home *Site Profile Agent* at Site 1, which now becomes the callee for this negotiation. It is assumed that a user cannot logon to two sites simultaneously. If such a situation occurs, the system reports an error and neither logons are accepted. However, if at any point, user logs on to two different sites consecutively, the last logon supersedes a previous logon. Thus, the current location is determined by the most recent logon. In the light of these assumptions, consider a hypothetical case in which a user is visiting two sites en route. After logon at the first visited site, the user may decide to logon to the second visited site, which is treated as a time-based logon or pre-logon. Further, multiple logons are accepted only for sequential time-slots. The second logon request is passed on to the *User Agent* of the user. This agent delays the logon in case the user may change his/her mind. Nevertheless, it activates the logon request just before the start-time of the second visit. This would require additional features to be included on *Site Logon Interface*; however, it is not supported by the implementation work in this thesis.

Consider that user 'A' roams into Site 2, depicted as a dotted oval at Site 2 (figure 5.6 above). Assuming, that she logs on through the foreign *Site Logon Interface*, (the one associated to Site 2). Thus, foreign *Site Profile Agent* is invoked and it contacts the home *Site Profile Agent* of the user A. The guest profile obtained at the end of negotiation contains the distinguish name(s) either of the devices or *Site Service Gateway* defined in section. This profile is either displayed at the time of logon or sent (e.g., by e-mail) for user's information.

Figure 5.7 shows a simplified state diagram. It does not show detailed information sent in the content of messages. It also does not depict error or exception recovery rules arising from NACK, communication or execution failure; these issues are not addressed here. Further, the language and ontology attributes are not included, since it is assumed that the agents are already aware of these, as they are interacting in a co-operative environment. The performatives used in this scenario are ask-if, tell, and untell.

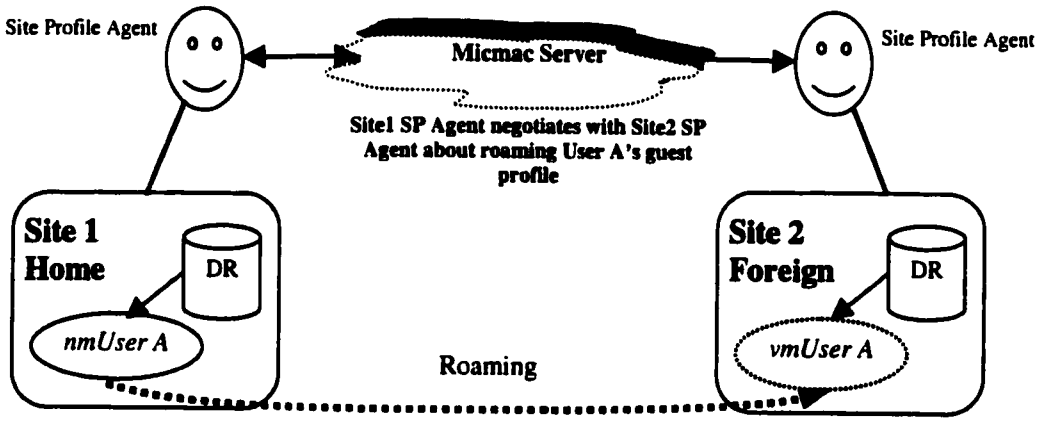


Figure 5.6: Home and Foreign Site Profile Agent Negotiation

In the first message of the foreign *Site Profile Agent*, it informs the home *Site Profile Agent* that they are going to start a conversation called 'Negotiate'. In the same message, inside the KQML performative 'ask-if', it authenticates itself to the home *Site Profile Agent.* Depending on the authentication id submitted, which could produce ACK or NACK at home *Site Profile Agent.* Assuming, that the ACK is produced, it is sent to Site 2 using KQML the performative 'tell'. Then the Site 2 sends 'sign-up' data to the Site 1, containing the logon information (e.g., e-mail id, user authentication, intended workspace area, user privilege) of the user 'A'. It should be noted that the npasUserlevel data is not sent at the time of user logon; this data is retrieved through the user's home

Site Profile Agent during sign-up. This npasUserlevel data must be understood across domains, which is possible by establishing organization-level service subscription contract that signifies co-operative networks; section 4.5.2.1 gives the definition of organization-level service contract. This sign-up information could also produce ACK or NACK, subject to the correctness of the user identity data submitted during logon. If ACK is the outcome, then the conversation goes into 'Profile Status', where it waits to receive one of the three results: 'NoProfile', 'Modify', or 'Oldfound', and the *Services Option Tree*, which is not indicated in figure 5.7 (a). for 'NoProfile' indicates that Site 2 does not have *vmUser* profile of user 'A' in its *DR*. 'Modify' indicates that Site 2 has found *vmUser* profile of user 'A' in its *DR*, and is not able to activate the old profile. Perhaps due to the change in the resource usage load at Site 2, or change in the *Site-related* data of the Site 2, etc. However, it is willing to negotiate. 'Oldfound', indicating that Site 2 has old *vmUser* profile of the user 'A', and is ready to activate it.

If 'NoProfile' is sent from Site 2, then Site 1 goes into 'New Profile' state, where it verifies several conditions of organization-level service subscription contract for a visitor. Since, organization-level service subscription contract is not specified yet, the details of 'New Profile' remains a future work. Subsequently, the conversation falls into the 'Negotiation' state. The 'Negotiation' state is decomposed in figure 5.7 (b) as two states: 'Process Services Option Tree' and 'GetApproved Profile'. The 'Process Services Option Tree' state receives the *Services Option Tree* from the previous state. The *Services Option Tree* is a tree-based data structure that stores string data and their inter-dependencies. The data in *Services Option Tree* contains information on type of services available at Site 2, and their interdependencies. The interdependence is used in the sense

that if one service is chosen from a node at a certain level then other services can or cannot be selected from that level. The home *Site Profile Agent* traverses through the *Services Option Tree* and chooses the options desired in view of the user 'A' *nmUser* profile. Once the *Services Option Tree* is processed, it sends back the 'FormedProfile', for the approval and the Site 2 responds with 'Accept'. This brings the negotiation to the final state: 'Done', as shown in figure 5.7(a). The *Services Option Tree* plays an important role in the agent negotiation. It reduces the overhead of messages, simplifying the conversation. However, this requires that foreign and home *Site Profile Agent* be complex enough to construct and process *Services Option Tree*. This also implies that *Services Option Tree* can not be sent as a very large chunk of data; its details appear in the section 5.4.1.

During the 'Process Services Option Tree' state, when it is found that options in *Services Option Tree* are insufficient, the 'OptionsInsufficient' is sent to the Site 2. By insufficient it is meant that devices/resources indicated in the attribute type reqDevicelist and desDevicelist cannot be met based on options provided in the *Services Option Tree*. Site 2 may or may not send another *Services Option Tree*, no transition takes place until Site 2 either sends *Services Option Tree* again or 'NotPossible'. This loop runs for once only, even if the profile request is not fulfilled. Therefrom, Site 1 would send the "FormedProfile" along with a request to upgrade the profile during the user's stay at the visited site, indicating the acceptance of whatever is provided from Site 2. This is the limitation of the negotiation strategy proposed in this work. In figure 5.7 (b), message ids are not indicated for the case when 'OptionsInsufficient' message is part of the conversation.

In the case when 'Modify' is received from the foreign *Site Profile Agent*, the scenario remains the same as for 'NoProfile' except that instead of going through 'New Profile' state the conversation enters the 'Modification' state and then goes into the 'Negotiation' state. In 'Modification' state, the specified or unspecified reasons, stating the inability to support the requested current profile, are noted.

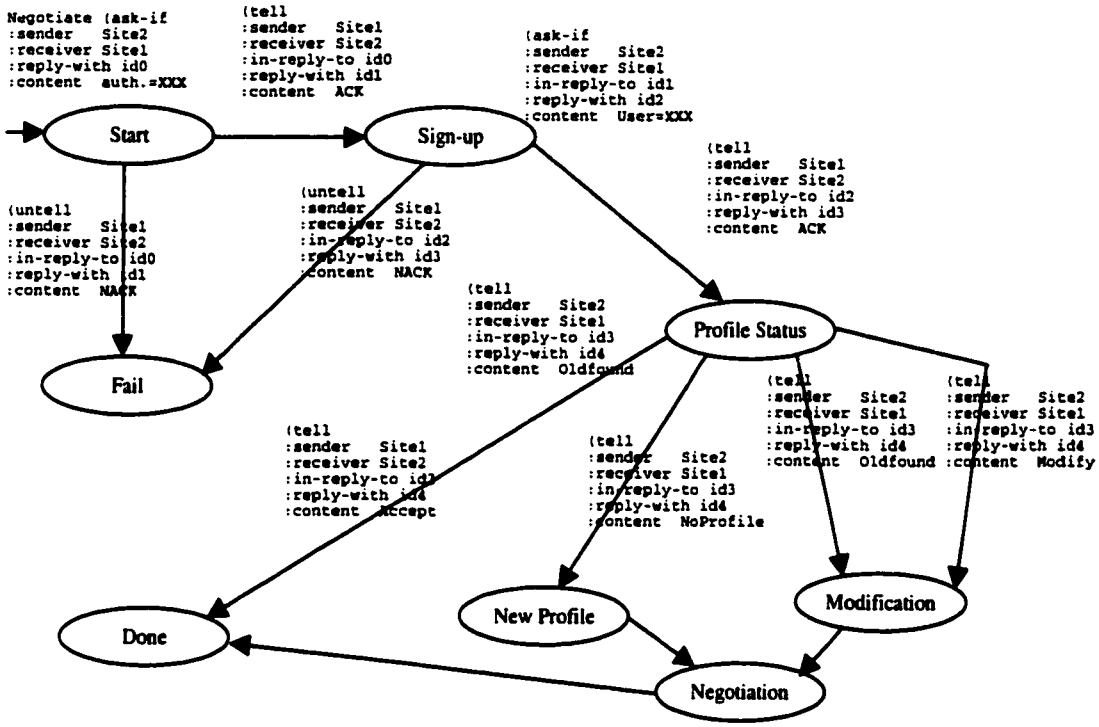


Figure 5.7 (a): Callee Conversation (Home Site Profile Agent at Site 1)

Finally, when 'Oldfound' is received from foreign *Site Profile Agent*, the home *Site Profile Agent* has two options. It can choose to either send 'Accept' to end negotiation or go into 'Modification' state. The home *Site Profile Agent* would accept the old profile, when it believes that user's current needs are matched by the old profile. On

contrary, if it finds that the old profile does not completely reflect the current needs of the user, it would fall into 'Modification' state and follow through the same scenario as for 'NoProfile'.

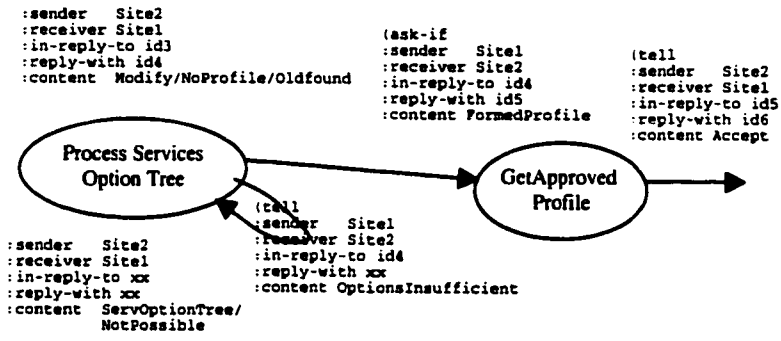


Figure 5.7 (b): Sub-states of Negotiation State of 5.7 (a)

5.4.1 Services Option Tree

The *Services Option Tree* is designed to reflect a multiple-choice questioner that allows a customer to select heterogeneous communication services in a controlled visitor environment. With each choice there might be interdependencies or constraints, describing the relation between options. This questioner is then formed into a hierarchical data structure called here as *Services Option Tree*, which itself may represent a collection of sub-trees, described shortly. Example of a questioner is illustrated in figure 5.8.

FixedPhone	☒LocalOnly	☐Domestic	☐International
		No Mobile	No Mobile
MobilePhone	☒Mobile		

Figure 5.8: A Services Option Tree is a Questioner

The visited site in conformance with user privilege, say npasVisit 4, and profile request generates the form shown in figure 5.8. In this form, rsrcDefintion is informally depicted as 'LocalOnly', 'Domestic', etc. Similarly, constraints are indicated as 'No Mobile' under each rsrcDefinition. The checked item indicates the selected options.

The representation of such a tree requires the use of attribute of type rsrcDefinition, which is defined above in the data model and a common language to describe constraints. A common language provides the basis to define constraints on services, which would be understood across organization. The following BNF representation incorporates these ideas, highlighting the attributes for generating a *Services Option Tree.*

```
Services OptionTree     =     npasUserlevel ["$" siteLimitation "$"
                              constraints] "$" OptionsForm ; definition
                              of siteLimitation and constraints are
                              given at the end of this BNF.
OptionsForm             =     [constraints] 1*(rsrcDefintion ["+"
                              constraints])
```

The attribute siteLimitation is not defined in the *NPAS* data model; however, it fits under npasWorkspaceorEquivalent object class. This attribute gives the reason(s) for inability to satisfy the profile request partially. It is a dynamically changing attribute; e.g., higher resource load, network device failures, and network upgrade. *Site Resource Manager* should puts its value into *Data Repository*. Similar to constraints this attribute should be defined using the common language. However, this work does not provide concrete representations of siteLimitation or constraint attribute types.

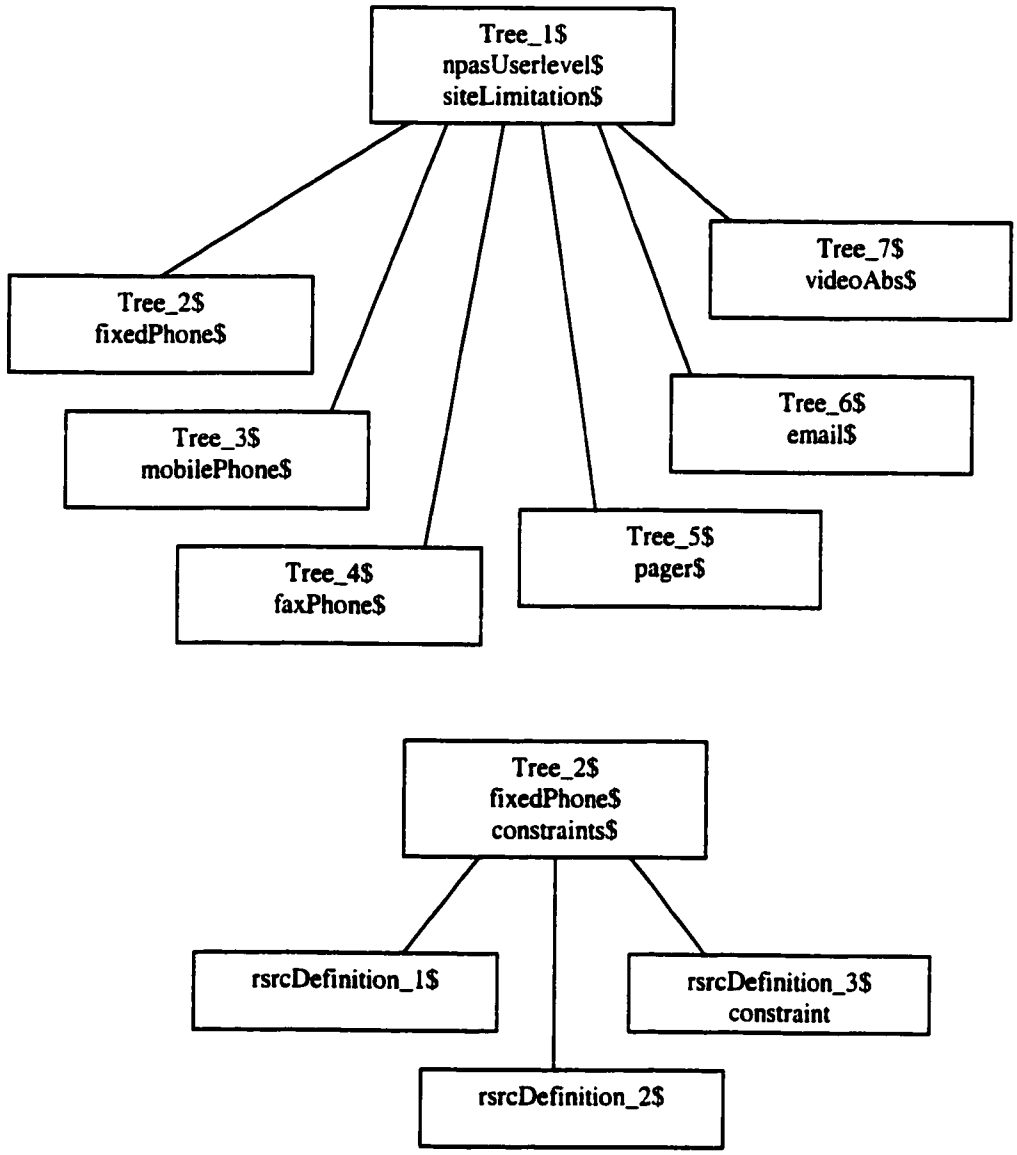


Figure 5.9: A Services Option Tree

Now based on the parameters indicated in BNF, the structure of *Services Option Tree* is illustrated in figure 5.9. According to figure 5.9, a *Services Option Tree* may consist of more than one sub-tree. It particularly is warranted when a visited site provides many options that may be associated to constraints, tending to produce a huge *Services Option Tree* and poses high bandwidth requirement. Although, billing was cited as the

part of *Services Option Tree*, it is considered beyond the scope of this thesis and is not discussed here.

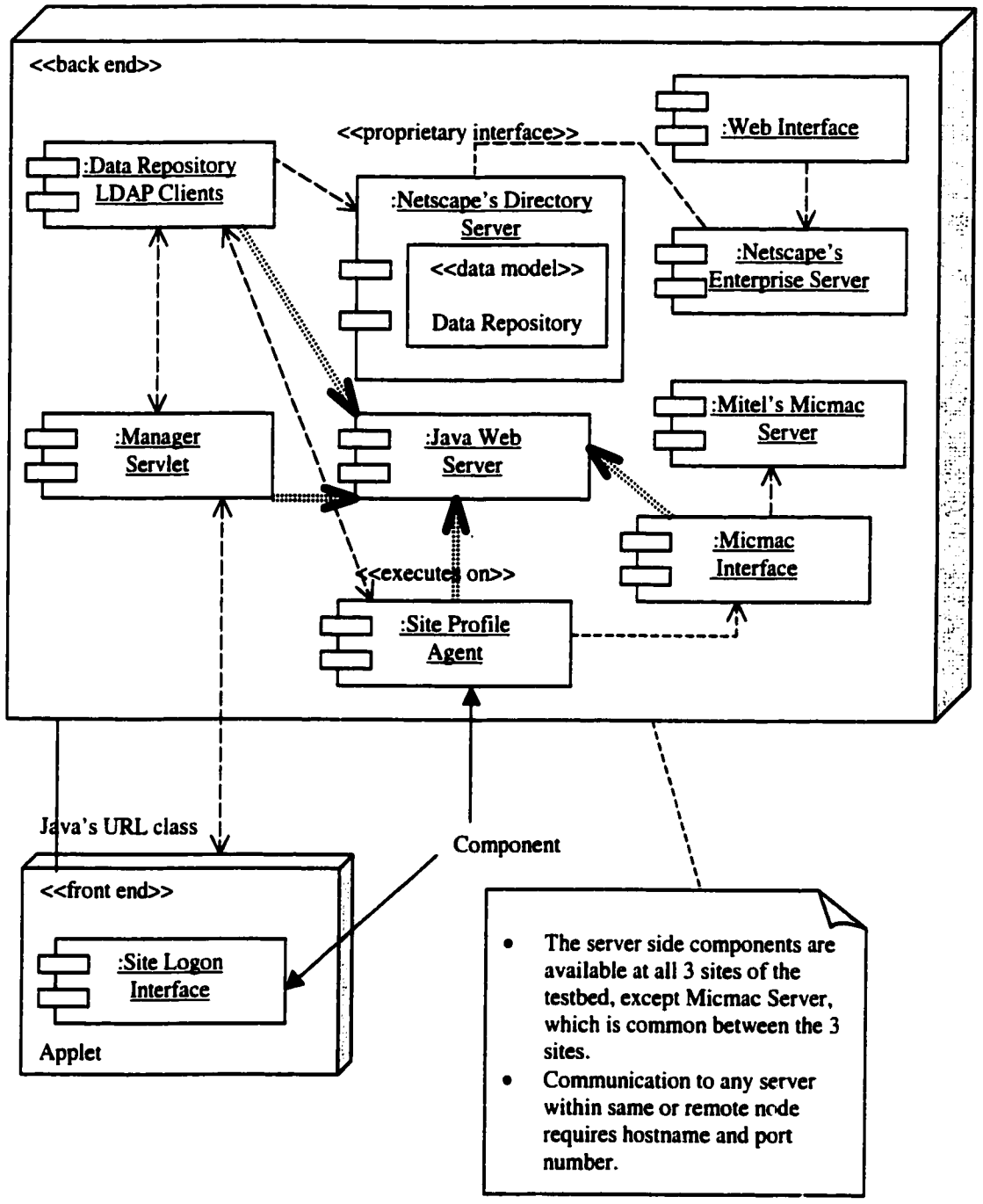


Figure 5.10: The UML's Deployment Diagram

5.5 Deployment, Package, Class and Interaction Diagrams

The UML figures allow explaining and summarizing the software design and implementation of the mobility data management components that are discussed above.

Figure 5.10 shows the UML deployment diagram for the mobility management components. It stereotypes components of mobility management as 'back end' and 'front end'. The back end components are not visible to user and they reside on a different computer in relation to a user computer that hosts front-end component(s), which is *Site Logon Interface*. Further, the figure uses a stereotype 'executes on' for those components that run on the virtual machine of the Java Web Server. Primarily, these components are implemented as servlets using Java Servlet API. LDAP clients are implemented using Java LDAP API, which are used by those servlets to query on *Data Repository*. The *Data Repository* is implemented on Netscape's Directory Server. To indicate that *Data Repository* encapsulates the data model, the stereotype 'data model' is created. The connection between the nodes containing front end and back end software are indicated to have communication channel established using Java's URL class. Finally, the note in the figure provides some explanation on distribution and communication.

The next diagram, figure 5.11, shows the package view of the code written for the components illustrated in figure 5.10. The package diagram shows distinction between application-based (e.g., LDAP Client, Site Profile Agent, etc.) and external packages (e.g., Servlet API, LDAP API, etc.) by creating a stereotype 'application'. The classes in the Manager Servlet and Site Profile Agent package are coded as servlets. Moreover it should be highlighted that internally packages are layered by developing a set of reusable classes that are used to code Manager Servlet and *Site Profile Agent*. It fulfills one of the

goals outlined for *Intermediate Objects*. Another application-based package is LDAP Clients that groups all the classes that allow adding, modifying, deleting, and searching *Data Repository*. The Micmac Interface package is used by the classes in *Site Profile Agent* package to communicate through Micmac Server. This shows that *Site Profile Agent* can only communicate through Micmac; however, if generalized interface were created, say Communication Interface, it would extend the ability to communicate through other means, e.g., RMI, etc. The application-based packages are named as (i) mamin.uottawa; (ii) mamin.uottawa.ldap; (iii) mamin.uottawa.Manager; (iv) mamin.uottawa.SPAgent; and (v) mamin.uottawa.MicmacInterface.

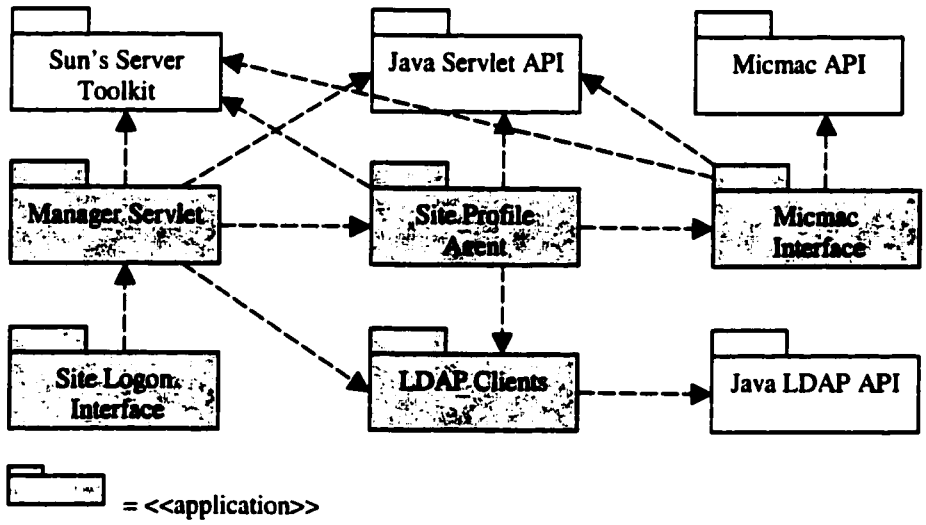


Figure 5.11: UML's Package Diagram

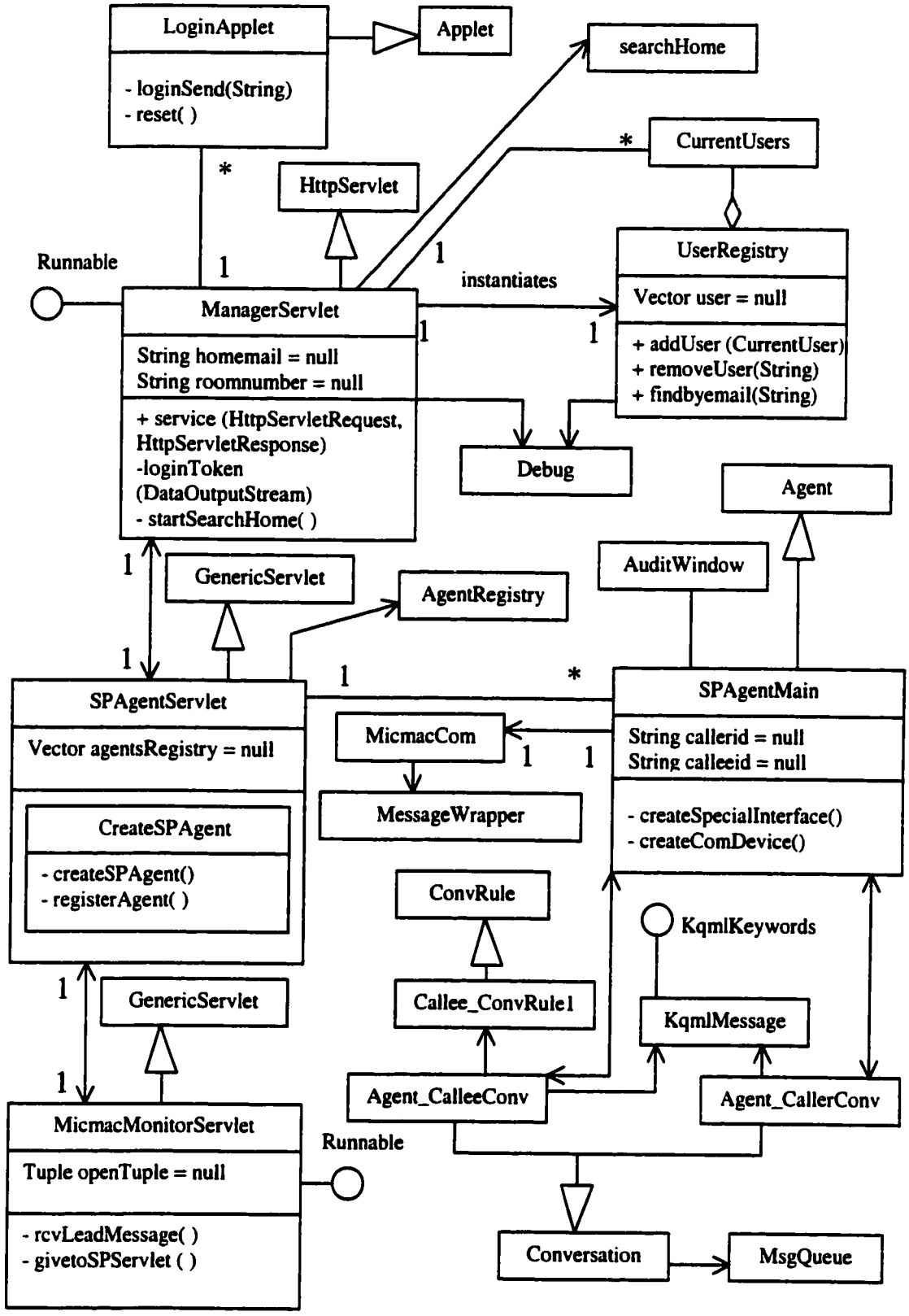


Figure 5.12: UML's Class Diagram

The following figure 5.12, presents the class diagram that shows *Site Logon Interface* and those back end components that realize the logon and negotiation. Some of the classes shown in the figure 5.12, for example Conversation, ConvRule are adopted from Java Based Agent Framework for Multi-Agent System (JAFMAS). Further, KqmlMessage and KqmlKewords classes are adopted from Lockheed Martin's KQML API.

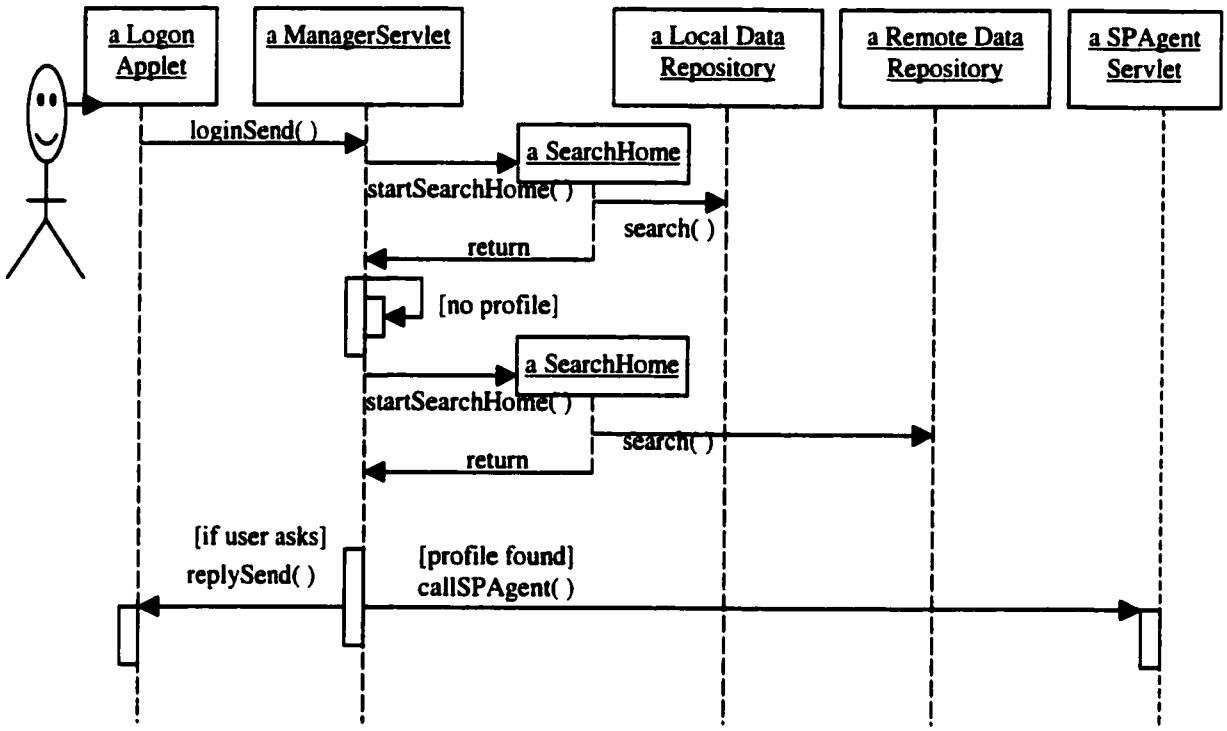


Figure 5.13: UML's Sequence Diagram for Logon and Agent Invocation

The UML's sequences diagram, figure 5.13 shows the dynamic view of those classes that interact to perform logon and invoke *Site Profile Agent*. The box line is used here to indicate self-delegation and concurrency rather than simple method activation. As per *NPAS's* LMT, if the logon is successful, then only *Site Profile Agent* is invoked. Therefore, here the figure 5.13 shows that when SearchHome – an LDAP client -

searches at local *Data Repository*, and if it fails then the search query is directed to the home *Data Repository* of the visiting mobile user. It is ManagerServlet that implements a simple algorithm based on the domain name in the email address to determine where is the *home Data Repostiory* of a visiting mobile use. The following is the pseudo code of the algorithm.

```
Always search in the Data Repository of the visited location first,
which is either of the three site-testbed

Else if home email address ends with 'uottawa.ca' then
      Search at home Data Repository of uottawa.ca

Else if home email address ends with 'nrc.ca' then
      Search at home Data Repository of nrc.ca

Else if home email address ends with 'mitel.ca' then
      Search at home Data Repository of mitel.ca
```

The figure 5.14 shows the dynamic view depicting the sequence of instantiations performed in the creation of caller conversation of *Site Profile Agent*. The constructor of SPAgentMain calls the constructor of its parent Agent class, which polymorphically invokes createComDevice in the SPAgentMain.

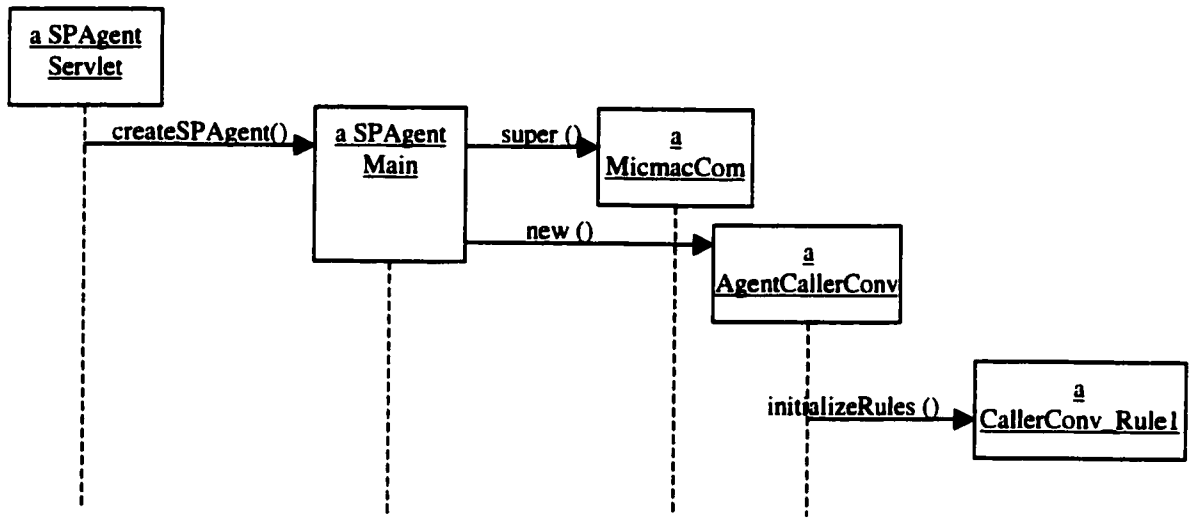


Figure 5.14: UML's Sequence diagram for Caller Conversation

The Java Servlet API is multithreaded. Therefore, conversations generated do not require asynchronous operation. When the *Site Profile Agent* is acting as a caller, the SPAgentServlet creates an instance of the caller conversation in a new thread. In addition, there is a server-like thread called MicmacMontiorServlet that runs continuously to monitor messages that are posted on Micmac for a new callee conversation. In this case, it will pass the request to SPAgentServlet to create instance of a callee conversation. Each caller and callee once instantiated create another MicmacCom object to pick and post KQML tuples on Micmac.

During implementation it was observed that Java Servlet API are sufficient to write servlets only; however, to communicate between servlets, Java Server tool kit API is required. Since, servlet-chaining mechanism of Java Servlet API does not provide flexibility to call other servlets arbitrarily.

The platform used for software development is Windows NT4.0. The coding is done using JDK1.1.3-5, Java Servlet Development Kit (JSDK) 1.0.1, JavaServer ToolKit 1.1, Java LDAP API 1.0 beta, and Java Micmac API. Netscape's Directory Server 3.0 b2 is being used as a stand-alone LDAP server, where the user and workspace-related X.500 object classes are stored and the sample data is stored. The Java Web Server 1.0.2 is used for Servlet execution environment. The Micmac Server provided by Mitel Corporation serves as a message transport medium for inter-agent communication.

Chapter 6

6 Conclusions and Future Work

In the context of nomadic computing, the problem of personal mobility has been investigated in this thesis. After, covering the background material on personal mobility and related work on the contemporary approaches to solve this problem, the idea of *Nomad's Personal Access System* (*NPAS*) has been presented. The thesis has identified and described the data management requirements for achieving personal mobility. The solution presented for data management has allowed exploring the various interesting threads of this problem: intelligent forwarding via home, dynamic mapping of user to devices using e-mail id, inter-site profile negotiation, issues pertaining to security policies of organizational resource usage, location management technique, among others. In addition, to realize the various artifacts of the design, further choices have been made. For example, LDAP and KQML have been selected to implement the data model and to accomplish the negotiation between agents, respectively. The analysis and some design details of other components, that are relevant to data management, have been presented, as well.

The thesis has offered a full scope functional understanding of data management aspect of personal mobility, and has expanded upon the idea of dynamic mapping of users to devices by involving the inter-site negotiation entity that respects the security concerns of shared environment. The development of the data model, namely *User, Workspace and Site-related* data in this thesis has sufficiently captured the functional

requirements of dynamic mapping of users to devices. The concept and design of *Site Profile Agent* proposed in this work provided the use of this data model.

The deployment of LDAP has entailed some qualitative benefits for the implementation of the data model; however, it remains an open problem to evaluate its performance in fully operational *NPAS* testbed. Since, dynamic data requirements of *NPAS*'s LMT may require more efficient persistent storage than a directory can provide. The notion of *Site Profile Agents* represents a step in the direction of achieving secure shared environment needed for supporting personal mobility. However, the issues pertaining to the security of the shared environment (*Site-related* data) has brought forward a strong need to develop a framework of organizational policies that can be used during the decision of profile assignment. This framework would also allow meeting the requirements of resource negotiation during a messaging session. Associated to this problem, is the negotiation itself that requires more investigation so that security concerns can be insured in a more granular and robust manner.

7 References

[1] O. Spaniol, *et al.*, "Impacts of Mobility on Telecommunication and Data Communication Networks", IEEE Personal Communications, 2(5), pp. 20-33, Oct. 1995

[2] R. Bagrodia, *et al.*, "Vision, Issues and Architecture for Nomadic Computing", IEEE Personal Communications, 2(6), pp.14-27, Dec. 1995

[3] M. Zaid, "Personal Mobility in PCS", IEEE Personal Communications, 1(4), pp.12-16, 1994

[4] M. Satyanarayanan, "Mobile Information Access", IEEE Personal Communications, 3(1), pp.26-33, Feb. 96

[5] J.Ioannidis, D.Duchamp and G.Q.Maguire Jr., "IP-based Protocols for Mobile Internetworking", Proc. of SIGCOMM'91, pp.235-245, Sept. 1991

[6] A. Myles, D.B. Johnson, and C. Perkins, "A Mobile Host Protocol Supporting Route Optimization and Authentication", IEEE Journal of Selected Areas in Communications, 13(5), pp.839-849, June 1995

[7] Corporation for National Research Initiatives June 1995, Cross-Industry Working Team – XIWT of Dept. of Computer Science, UCLA, "Nomadicity in the NII", http://www.xiwt.org/documents/Nomads_doc/NomadsTOC.html

[8] C.G. Harrison, "Smart Networks and Intelligent Agents", Proc. of Mediacom'95, Southampton, UK, Apr. 11, 1995

[9] A.Dearle, "Towards Ubiquitous Environments for Mobile Users", IEEE Internet Computing, 2(1), pp.22-32,1998

[10] T. Richardson *et al*, "Virtual Network Computing", IEEE Internet Computing, 2(1), pp.33-38, 1998

[11] B.Schilit, N. Adams, R. Want, "Context-Aware Computing Applications", Proc. of IEEE Mobile Computing Systems and Applications Workshop, Los Alamitos, CA, USA, 1995

[12] T. Imielinski and B.R.Badrinath, "Mobile Wireless Computing: Solutions, and Challenges in Data Management", Technical Report DCS-TR-296/ WINLAB-TR-49, Dept. of Computer Science, Rutgers University, New Brunswick, NJ

[13] B. Bruegge and B. Bennington, "Applications of Mobile Computing and Communications", IEEE Personal Communications, 3(1), pp.64-71, Feb. 1996

[14] P. Nixon and V. Cahill, " Mobile Computing: Technologies for a Disconnected Society", IEEE Internet Computing, 2(1), pp.19-21,1998

[15] R. Pandya, "Emerging Mobile and Personal Communication Systems", IEEE Communications Magazine, 33(6), pp.44-52, 1995

[16] M. Voelker and B. N. Bershad. "Mobisaic, An Information System for a Mobile Wireless Computing Environment", Proc. of IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US, Dec. 1994

[17] A.Grinberg, "Seamless Networks: Interoperating Wireless and Wireline Networks", McGraw Hill, 1997

[18] Computer Science and Telecommunications Board, "Realizing the Information Future: The Internet and Beyond", National Academy Press, Washington, DC. 1994

[19] M. R. Genesereth and S. P. Ketchpel, "Software Agents", Communication of the ACM, 37(7), pp.48-54,1994

[20] P. Domel, *et al.*, "Mobile Agent Interaction in Heterogeneous Environments", Lecture Notes in Computer Science 1219, pp. 136-148, Springer Verlag, Berlin, Germany, 1997

[21] T.Finin, *et al.*, "Mobile Agents Can Benefit from Standards Efforts on Interagent Communications", IEEE Communications Magazine, 36(7), pp.50-56, July 98

[22] D. Chess, *et al.*, "Itinerant Agents for Mobile Computing", IEEE Personal Communications, 2(5), pp.34-49, Oct, 1995

[23] Y. Labrou, "Semantics for an Agent Communication Language", Ph.D Thesis, Computer Science Department, University of Maryland, Baltimore County, 1996

[24] Jan Jannik, *et al.*, "Efficient and Flexible Location Management Techniques for Wireless Communication Systems," Proc. of MOBICOM'96, Rye NY, Nov. 10-12, 1996

[25] T. Magedanz, "On the Impacts of Intelligent Agent Concepts on Future Telecommunication Environments", Proc. of 3rd Int. Conf. On Intelligence in Broadband Services and Networks, Germany, pp. 396-414,1995

[26] J.Z.Wang, "A Fully Distributed Location Registration Strategy for Universal Personal Communication Systems," IEEE Journal of Selected Area in Communications, 11(6), pp.850-860, Aug. 1993

[27] E.Buitenwerf, *et al.*, "UMTS: Fixed Network Issues and Design Options", IEEE Personal Communications, 2(1), pp.30-37. Feb. 1995

[28] K. Chiang, *et al.*, "Mobility Supported by IN in UMTS/B-ISDN", Proc. of IEEE Int. Conf. On Information, Communications and Signal Processing, Singapore, Sept. 9-12 1997

[29] G. Alfano and D. Ferro, "IN and CTM Applications: Reality and Opporutnities", Proc. of 5th Int. Conf. on Intelligence in Networks, Bordeaux, France, May 13-15, 1998

[30] N. Faggion, "Intelligent Networks in Mobile Communications", Proc. of 5th Int. Conf. on Intelligence in Networks, Bordeaux, France, May 13-15, 1998

[31] M. Laitinen, *et al.*, "Integration of Intelligent Network Services into Future GSM Networks", IEEE Communication Magazine, June 1995

[32] I. Iido, *et al.*, "DUET: An Agent-Based Personal Communications Network", IEEE Communications Magazine, 33(11), pp.44-49, Nov.1995

[33] T. Eckardt, *et al.*, "A Personal Communication Support System based on X.500 and X.700 standards", Computer Communications, 20(3), pp. 145-156, May 1997

[34] J. Bradshaw (ed.), "Software Agents", AAAIPress/The MIT Press, 1997

[35] N. Portex, *et al.*, "VPN and Mobility: An Innovative IN Feature set Generating new Revenue in Deregulated Network", Proc. of 5th Int. Conf. on Intelligence in Networks, Bordeaux, France, May 13-15, 1998

[36] S. Abu-Hakima, *et al.*, "A Multi-Agent System for Seamless Messaging by Email, Fax or Voice Mail," Proc. of Int. Joint Conf. on AI, IJCAI'97 Workshop on AI in Distributed Information Networking, Negoya, Japan, Aug. 24, 1997, reference w4 to IJCAI, p.9-16. http://www.nrc.ca/iit/SPIN_public

[37] B. Awerbuch and D. Peleg, "Online Tracking of Mobile Host", JACM, 42(5), pp.1021-1058, Sep. 1995

[38] M. Fowler and Kendall Scott, "UML Distilled: Applying the Standard Object Modeling Language", Addison-Wesley, 1997

[39] P.Krishna, "Performance Issues in Mobile Wireless Networks," Ph.D thesis, Texas A&M University, Aug. 1996

[40] D.C.C.Wang, "A Survey of Number of Mobility Techniques for PCS," Proc. of Third Int. Conf. on Universal Personal Communications, ICUPC'94, New York, USA, 1994

[41] Henning M. *et al.*, "Data Management for Wide-Area Mobility in Private Telecommunication Networks", ITG-FACHBERICHTE, No.124, pp. 535-546, 1994

[42] V. K. Garg, *et al.*, "Subscriber Data Management in Personal Communications Services", IEEE Personal Communications Magazine, 4(3), pp.33-39, June 1997

[43] H. Maab, *et al.*, "Directory Services for Mobility Management in Private Telecommunication Networks", Proc. of IEEE Int. Conf. on Communications, Piscatway, NJ, 1993

[44] G. Havermans and W. Pasman, "Mobility in Private Networks", Philips Telecommunications Review, 51(2), pp.35-40, 1993

[45] ITU-T Recommendation X.500 (1993) | ISO/IEC 9594-1 :1993, Information technology – Open Systems Interconnection, "The Directory: Overview of Concepts, Models, Services", http://www.dante.net/np/ds/osi.html, 1993

[46] W. Yeong, T. Howes, S. Kille, "Lightweight Directory Access Protocol", RFC 1777, March 1995. W. Yeong, T. Howes, and S. Kille, "Lightweight Directory Access Protocol", RFC 1777, ftp://ds.internic.net/rfc/rfc1777.txt , Mar. 1995

[47] M. Wahl, T. Howes, S. Kille, "Lightweight Directory Access Protocol (v3)," RFC 2251, December 1997, ftp://ds.internic.net/rfc/rfc2251.txt

[48] L. A. G. Oliveria, *et al.*, "An Agent-Based Approach for Quality of Service Negotiation and Management in Distributed Multimedia Systems", First International Workshop, Mobile Agents (MA) '97 Proceedings. Springer-Verlag, Berlin, Germany; pp.1-12, 1997

[49] A. Reinhardt, "The Network with Smarts", Byte, pp.51-64, Oct. 1994

[50] J. Hartman, "The Alignment of IN and GSM", Proc. of 5th Int. Conf. on Intelligence in Networks, Bordeaux, France, May 13-15, 1998

[51] G. Irvine, *et al.*, "A Futuristic Service Architecture for Personal Interactive Communications", Proc. of 7th IEEE Intelligent Network Workshop, Bordeaux, France, May 10-13, 1998

[52] R. Ramjee, *et al.*, "The Use of Network-Based Migrating User Agents for Personal Communication Services", IEEE-Personal-Communications, 2 (6), pp. 62-68, 1995

[53] H. Velthuijsen and N. Griffeth, "Negotiation in Telecommunications Systems", Proc. of AAAI Workshop on Coordination Among Heterogeneous Intelligent Systems, 1992, San Jose, CA

[54] C. J. Petrie, "Agent-Based Engineering, the Web, and Intelligence", IEEE Expert, 11 (6), pp. 24-29, 1996

[55] M. Rizzo and I. A. Utting, "An Agent-based Model for the Provision of Advanced Telecommunications Services", Proc. of TINA'95 Conf, Melbourne, Australia, Feb 13-16, 1995

[56] R.Orfali, *et al.*, "Client/Server Programming with Java and CORBA", John Wiley & Sons, Inc., 1997

[57] T. Howes and M. Smith, "Programming Directory-Enabled Applications with Lightweight Directory Access Protocol", Macmillan Technical Publishing, USA, 1997

[58] D. Chadwick, "Understanding X.500 – The Directory", International Thomson Computer Press, UK, 1996

[59] M. Ahmed, A. Karmouch, S. Abu-Hakima, "MediaABS: A Multi-Format Video Processing System", Submitted for Publication

[60] M.Whal, T. Howes, S. Kille, "LDAP(v3): Attribute Syntax Definitions", RFC 2252, Dec. 97, ftp://ftp.isi.edu/in-notes/rfc2256.txt

[61] M.Wahl, "A Summary of the X.500(96) User Schema for Use with LDAPv3", RFC 2256, Dec. 97, ftp://ftp.isi.edu/in-notes/rfc2256.txt

[62] D. H. Crocker, "Standard for the Format of ARPA Internet Text Messages", RFC 822, Aug. 13, 1982, http://info.internal.isi.edu:80/in-notes/rfc/files/rfc822.txt

[63] B. Falchuk, A. Karmouch, "The Mobile Agent Paradigm Meets Digital Document Technology - Designing For Autonomous Media Collection", Multimedia Tools and Applications, Vol.8, No.1, Kluwer Academic Publishers, January 1999

[64] Micmac Web Site, http://micmac.mitel.com

8 Publications

[1] A.Hooda, A.Karmouch, S.Abu-Hakima, "Personal Mobility Management using LDAP Distributed Directory and Static Agents", Proc. International Conf. on Intelligence in Networks, Bordeaux, France May 10-15, 1998

[2] A. Hooda, A. Karmouch, S. Abu-Hakima, "Nomadic Support Using Agents", Proc. 4th International Symposium on Internetworking, Ottawa, July 6-18, 1998

[3] A. Hooda, A. Karmouch, S. Abu-Hakima, "Agent Negotiation for Supporting Personal Mobility", Proc. 2nd Int. Workshop on Intelligent Agents for Telecommunications Applications (IATA'98), France, July 4-7, 1998. In Lecture Notes in Artificial Intelligence (1437), A. Albayark, F. Garijo (Eds.), pp.190-203

[4] A. Hooda, A. Karmouch, S. Abu-Hakima, "Managing Mobility of Users in a Virtual Network", to appear in Multimedia Tools and Applications, Kluwer Academic Publications.