Performance Evaluation of a Mobile Agent-Based Platform for Ubiquitous Service Provision

V. Baousis, M. Kyriakakos, S. Hadjiefthymiades and L. Merakos

University of Athens, Department of Informatics & Telecommunications,
Communication Networks Laboratory, Panepistimioupolis, Ilisia, Athens 157 84, Greece.

Email: **bbaous|miltos|shadj|merakos@di.uoa.gr**

ABSTRACT

The Virtual Home Environment (VHE) is a very important mechanism in the upcoming mobile telecommunications infrastructure, providing a ubiquitous approach in the delivery of services. The universality of systems like UMTS and WLANs increases the need for efficient introduction of VHE. In this paper, we discuss the adoption of Mobile Agents (MA) for coping with the VHE technical challenges. We present the detailed design of an integrated, open and extensible architecture by taking into account a variety of use cases and technical alternatives (mobile devices, fixed terminals). Relevant implementations are presented to indicate the feasibility of the discussed technical orientation. Finally, we study and compare the performance of the proposed agent based framework with a conventional solution.

Keywords: Service Management, Mobile Agent Technology, Virtual Home Environment, Mobile Computing.

1. INTRODUCTION

Nowadays the ability to access services and obtain information anywhere and anytime, irrespective of the network and terminal is imperative to meet the requirements of contemporary users. From the viewpoint of network operators and service providers, meeting these requirements is a challenging issue, since many access types and service technologies should be seamlessly combined in order to deliver advanced services to the end users. Moreover, network operators and service providers, should adapt to the new emerging technologies in a reasonable time in order to preserve their customers/users and be competitive in the market.

One of the service provisioning aspects in UMTS is the capability to provide to users services with the same look and feel, same interface capabilities, taking into account user's service specific preferences as if they were in their home network irrespective of the access network and device capabilities. This option has been addressed in the 3rd Generation Partnership Project (3GPP) as part of the service aspects in UMTS and is generally referred to as the Virtual Home Environment (VHE) [1]. VHE is defined as *"… a concept for personal service environment portability across network boundaries and between terminals"*. The idea of VHE is to offer a "service" to every user in any network and provide all kinds of services in all types of terminals while considering the user preferences.

In this paper we present a framework for the realisation of VHE using Mobile Agent (MA) technology which is a promising technology for communicating and managing functional components of a mobile service. An agent is a software entity that accomplishes tasks on behalf of its owner, user or some other software entity, with some degree of independence or autonomy. Agents may be autonomous (i.e., pursue a certain task independently of their user), proactive, distributed, cooperative, self-learning and adaptive. Moreover, MAs are free to travel through the network by transporting themselves (i.e., their execution state, code and data) from node to node. This feature allows a mobile user to perform various tasks even if his/her device is temporarily disconnected.

In the past, several studies ([13], [14], [15]) have tried to assess the performance of the MA paradigm against the client-server and the remote invocation paradigms. However, agents are not trying to replace traditional ways of communication but to enhance the functionality and operation of the involved service entities. Another issue that influences the performance of a MA platform is their execution environment. There are tens of such agent platforms [25] and most of them have been implemented for specific applications. In our VHE architecture, we adopted the Grasshopper [25] MA platform.

A detailed presentation of an integrated, open and extensible architecture is provided by taking into account a variety of use scenarios and technical options (i.e., mobile devices, fixed terminals). Furthermore, the performance of the proposed MA-based framework is studied and compared with a conventional (remote invocation) solution. Finally, relevant implementations and research activities are discussed.

The rest of the paper is structured as follows. Section 2 discusses background knowledge on VHE technologies while Section 3 presents relevant previous work. In Section 4, we present the proposed MA framework while in Section 5 we describe a performance assessment model for the proposed architecture. In Section 6 performance evaluation results are presented. Finally, conclusions and directions for further work are briefly presented in Section 7.

2. BACKGROUND KNOWLEDGE

The VHE should be regarded as an umbrella concept that covers many technologies pertaining to service provisioning. In this Section, we briefly describe all these technologies to set the scene for the study and improvement of the VHE.

2.1 Virtual Home Environment

According to the 3GPP VHE specifications, the VHE Business model and the user perspective are structured as shown in [1] (Figure 1 and Figure 2).

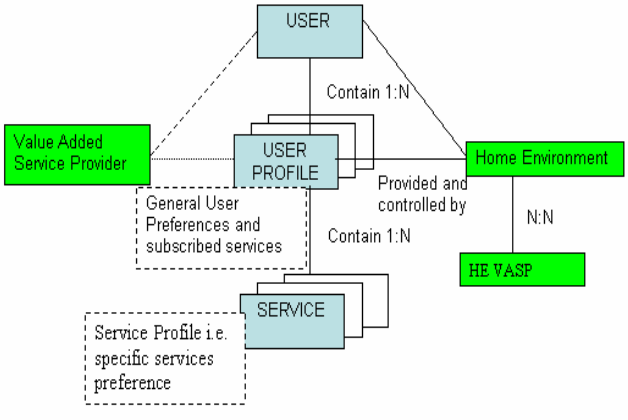

Figure 1 :VHE – Business Model

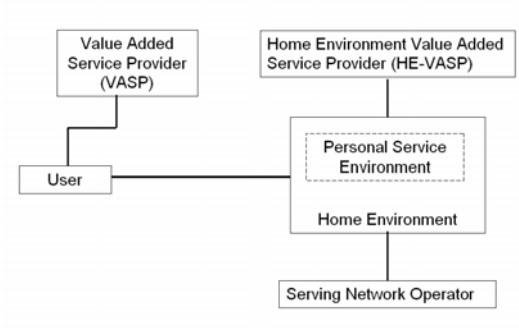

Figure 2: VHE – User's Perspective

The entities shown in Figure 1 and 2 have the following roles-functionalities:

- **User**: subscribes, accesses, customises and invokes services through the Home Environment and Value-Added Service Providers.
- **User Services Profile**: contains information on subscriber services, their status and service preferences. This is part of the User profile information, necessary to provide a user with a consistent, personalised service environment.
- **Services**: functionality or content provided to the user.
- **Home Environment (HE)**: transparently provides services to the user in a managed way, possibly by collaborating with Home Environment – Value Added Service Providers (HE-VASPs). HE also provides and controls the user's Personal Service Environment (PSE), ensures user's access in his/her PSE and provides mechanisms to support identical services provided by HE-VASPs when the user moves across network boundaries (network roaming) and switches between terminals.
- **Serving Network Operator**: Offers network connectivity, network-specific capabilities and services to the user via the HE. The serving network can be a Home Network (HN) or a Visited (roamed) Network.
- **Value Added Service Provider (VASP)**: The user may access services directly from VASPs. Services obtained directly from VASPs are not managed by HE and, therefore, are not part of the VHE service offered by the HE.
- **Home Environment – VASP (HE-VASP):** the user personalises and accesses the services provided by the HE-VASP through the HE. HE may allow HE-VASPs to access its service toolkits in the network (e.g., Open Service Access - OSA, Customized Application of Mobile network Enhanced Logic - CAMEL) and in the Mobile Equipment (e.g., Mobile Station Application Execution Environment - MeXE) for the execution of services provided by the HE-VASP.

2.2 Open Service Access (OSA)

OSA [2] is a flexible framework that enables applications to make use of network functionality which is defined as a set of Service Capability Features (SCFs) and are supported by different Service Capability Servers (SCS). The SCS serve as gateways between the network entities and the applications. Examples of SCFs are call-control, location/positioning and notification. The OSA API is based on protocols such as CORBA/IIOP, and SOAP/XML. OSA enables service providers to make use of network functionality such as call (session) control, management of conferences and location information. It also offers an interface for application authentication/authorization, discovery of service capabilities and user status (Profile, Terminal).

2.3 UMTS architecture

A typical UMTS architecture is shown in Figure 3. RNC (Radio Network Controller) is the heart of the UMTS Radio Access Network (RAN). RNC is linked to the packet-switched and

circuit-switched domain (the UMTS core network), and integrates a high-speed packet switch. Additionally, RNC is capable of supporting interconnections to other RNCs, managing radio resource and handling mobility management. RNC controls a large number of Node Bs which is the network entity that serves a single cell.

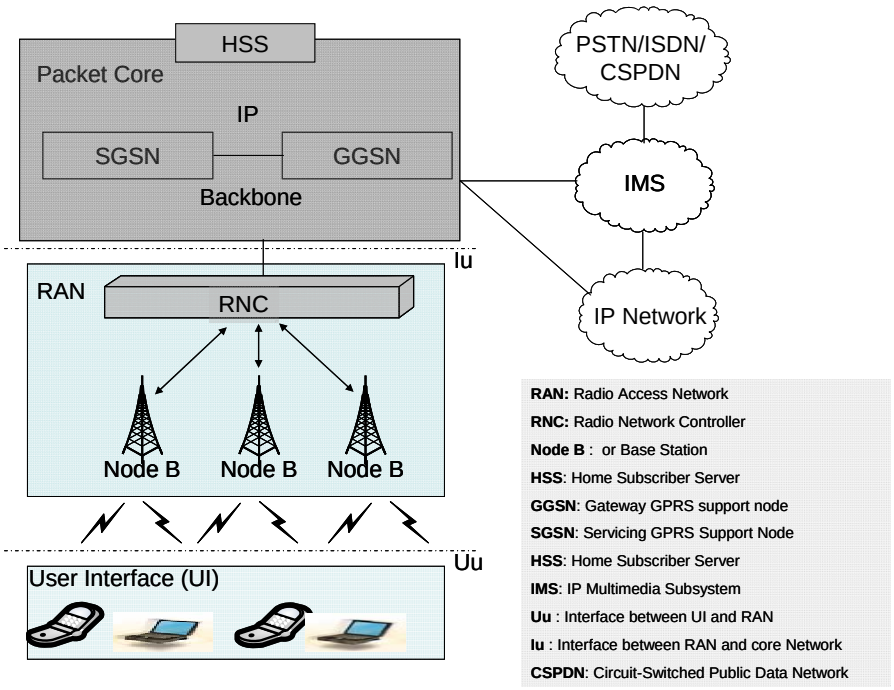

Figure 3: UMTS architecture (Release 6+)

3. PRIOR WORK ON THE VIRTUAL HOME ENVIRONMENT

In this Section we briefly present other research activities that investigate the VHE concept and discuss prototype implementations.

One of the most important contributions on the use of MA in telecommunication networks is the work in [3]. The architecture in [3] has influenced many other research proposals (including our own) that adopt MA. Mobile and stationary agents reside in a Distributed Agent Environment (DAE) that spans vertically the whole network from the user's device to the access and core network and to other 3rd party Service providers.

In [4] an agent based service provisioning approach is introduced for the realization of the VHE. An Adaptive Profile Manager (APM) is implemented offering advantages for flexible service provisioning, remote modification of user profiles for different services, management of different devices for service access and support of automatic download for updated services. The communication model is realised by exchanging messages between various agents, either stationary or mobile. Three services are supported by the APM: Intelligent Communication Manager, Personal Data Query and Information Retrieval.

The development of two sophisticated, agent-based telecommunication services, Adaptive Profile Manager (same as in previous paragraph) and Virtual Address Book (VAB) is discussed in [5]. VAB allows the user to access and modify his/her data in a consistent manner independent of location, access point and terminal equipment. Service realization is obtained through a layered agent-based model.

An agent based VHE architecture is proposed in [6]. The idea is to insert a middleware layer between the end user and a UMTS network for the support of VHE. Such middleware consists of a distributed agent environment built on top of CORBA.

A VHE, component-based architecture is investigated in the IST VESPER project [8]. Both the Network and Terminal sides have been considered and an API was implemented for accessing their functionality. A similar perspective is adopted in [7], where the collaboration between a home and a serving network is examined and a roaming contract between them is considered necessary for the operation of the VHE.

A MA based VHE approach is discussed in [9] where several agents (stationary, mobile) are cooperating for service and terminal portability. The proposed framework was not implemented or tested. A large number of MAs migrate from the same origin to the same destination by performing similar and very trivial tasks. It is unclear if there is any improvement on network performance. It is suggested that the MA that migrate to the user's terminal take into account the processing, persistence and memory capabilities of the mobile device. Nevertheless, this seems to be unfeasible or difficult to implement. In our framework, network utilization is of primary concern, thus the number of MAs used is the least possible and agents migrate only when it is necessary.

A different approach for the realization of the VHE is presented in [10]. Roaming users are represented through Personal agents with a unique identifier (SIP, URL) that enables them to participate and collaborate in communities. An interesting aspect is the adoption of the Resource Description Framework (RDF) for agent properties. By using semantics in the agent description, the lookup action of the other agents is performed transparently to the user. This is the earliest adoption of semantics in agents for the realization of the VHE. In our framework, MAs used for the realisation of VHE are user-bound, and service MAs migrate close to the end-user to offer their service. Service agents are cached locally and are available for future user requests.

The approach in [11] is to present roaming users with services provided by local providers. This would reduce extra network traffic, increase resource utilization and diminish security threats. Although the main characteristics of the VHE are maintained, service subscription and customization, user service profiles and home network control of the provided services are not considered. Formal semantics are exploited through ontologies for the description of the offered services and context. This approach bears an important disadvantage. The VHE platform will need to maintain a large amount of data for the service profiles and perform a best matching algorithm between the user's subscribed services and the available VHE services. Even though this approach deviates from the VHE and is not based on MA, it is included in this section in order to illustrate the trend of using formal semantics to unambiguous describe service functional components.

4. ARCHITECTURE DESCRIPTION

4.1 Framework description

In this section an overview of our MA-based VHE architecture is presented. In the proposed architecture, MAs contain and transport service logic or data from one network entity to another in order to realize the VHE. Stationary agents encapsulate the network functionality and collaborate with MAs. Such collaboration involves the provision of network resources, execution environment and the appropriate logic. A general description of the architecture is depicted in Figure 4. [16] describes the user interactions with the platform section and the possible VHE related requests a user may submit to the platform and how these requests are realised within the agent platform. Detailed UML diagrams have been used to depict the message exchange between agents.

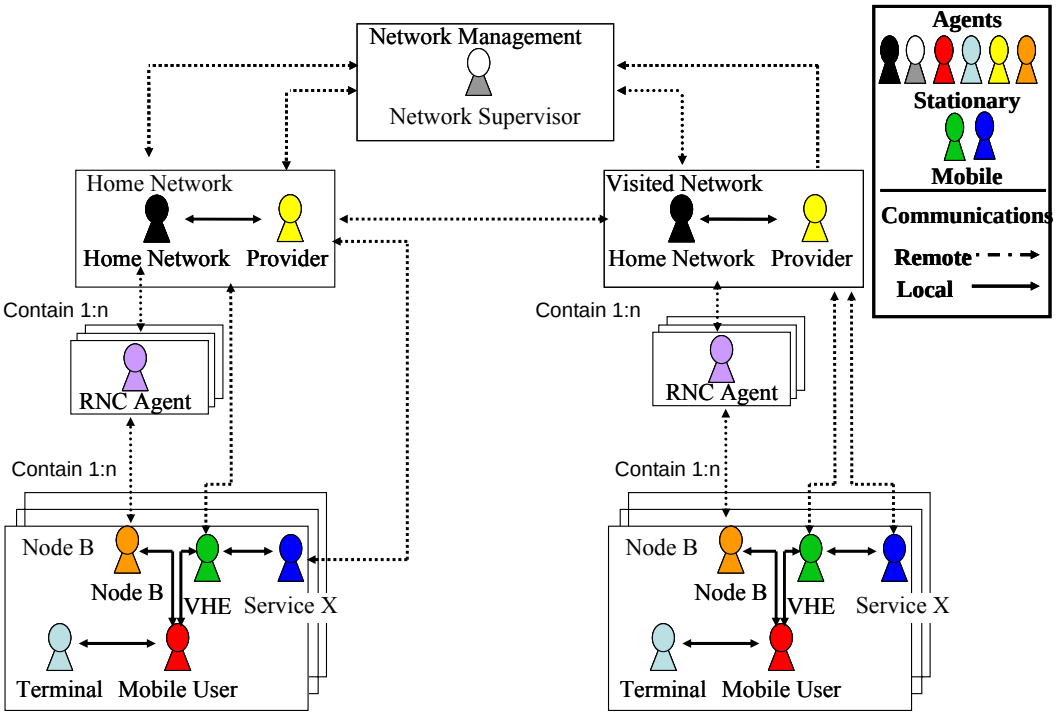

Figure 4: Mobile agent VHE architecture

The agents participating in the proposed architecture have the following functionalities and responsibilities:

Network management agents

- **Network Supervisor Agent (NSA)** is a stationary agent responsible for network interconnection and management. Associated networks have to register with NSA. Throughout this operation the NSA maintains a detailed list of the services offered by the collaborating networks. NSA also provides information on service availability. The role of the NSA could also be performed by any (properly enriched) Network Agent.
- **Network Agent (NA)** This entity is a stationary agent responsible for the network management and interconnection. Its main responsibilities include user authentication, and registration, supervision of the correct operation of RNC Agents and, finally, message routing, to and from other networks.
- **Provider agent (PA**) is a stationary agent that manages all supported services (Service Agents). The PA maintains an overview of all available services within the provider (network) domain. The PA is responsible to create the Service agents that contain the service logic of a service and dispatch them to the appropriate network entity, in order to provide a service to the requesting user. PA also creates VHE agents with user specific data (user service profile).
- **RNC Agent (RNCA)** is a stationary agent responsible for the interconnection of Node Bs and message routing from Node B Agents to Network management entities (Network and Provider agents) and vice-versa.
- **Node B Agent (NBA)** is a stationary agent responsible for Node B management and message routing for the users to the RNCA that the NBA belongs to and vice-versa.

VHE platform agents

- **Service Agent (SA)** is a mobile agent that represents an offered service. Depending on the service implementation scenario, the SA carries data, files, service logic or service results or all of them from the user's Home Network (HN) or a Service Provider to the visited network where the user currently roams.

- **Mobile User Agent (MUA)** represents the (mobile) user within the network.
- **VHE-Agent (VHEA)** is a mobile agent that is created by the PA and is dispatched to the place where the MUA is located. It contains information about user's service profile, such as personal data, preferences regarding subscribed services etc.
- **Terminal agent (TA**) is a stationary agent that allows the terminal to inform the network system about its capabilities (expressed using CC/PP).

4.1.1 Mobile agent structure

In general, a MA is used as a user's representative in the network and is capable of roaming, finding, executing services and delivering results to the user. A typical structure of a MA includes the following components: data state, migration policies, communication and code. The data state component contains the information carried by the MA during migrations; the migration policies component specifies the autonomous behaviour of the MA. The communication component is responsible for the MA's communication with other agents or network entities and finally the code implements the MA's autonomous behaviour with the support of the other three components.
Each component is configured according to the MA's task. Moreover, it should be noted that the Grasshopper platform provides a basic communication frame for the agents as well as general classes with the basic functionality needed either by stationary or by mobile agents. In the Table 1 an abstract structure of the MAs used for the realization of the VHE is shown.

Table 1: Mobile agent structure

	Data State	Code	Migration Policies	Communication
Mobile User Agent (MUA)	Contains a description of the terminal capabilities.	Methods for the mobile agent functionality and autonomous character	Created in the Node B where the user logons. Follows the roaming user to each new NBA.	Interfaces with the TA, NBA, VHEA and SA.
VHE Agent (VHEA)	Contains the User's Service Profile (includes information about the subscribed services, the provider for each service, service invocation parameters, user's preferences / customisations and configuration parameters that should be applied to the visited network in order to provide subscribed services).		Created by the user's PA and migrates to NBA. Follows the roaming user to each new NBA (is retracted by the MUA when the users move to other Node B or other network).	Interfaces with the PA, MUA and SA.
Service Agent (SA)	Contains the Service Data, Logic or Service Results.		Implements methods to autonomously migrate to a NA, to a RNC, or to a NBA.	Communicates with the NA, VHEA and the MUA.

In our architecture, MAs use OSA to communicate with stationary Agents (Figure 5). Stationary agents encapsulate the network functionality using OSA, and, thus, provide network resources, execution environment and the appropriate logic to MAs.

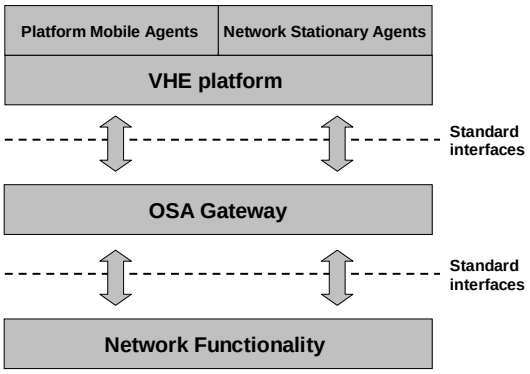

Figure 5: Conceptual Framework

The functionality of our agent-based VHE has been extended to cover mobile devices. We use Java 2 Micro Edition (J2ME) [12] that is integrated to the mobile agent platform. Agents reside only on the fixed network and dispatch the requests of the mobile user to the appropriate fixed network entities (Figure 6).

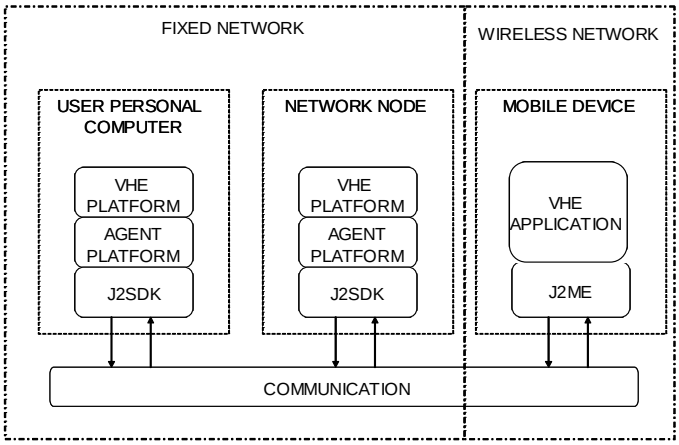

Figure 6: Network and Devices Communication

4.1.2 Framework implementation details

The proposed MA framework for VHE was developed in the Grasshopper. Grasshopper is built on top of a distributed processing environment, is compliant with the MASIF standard [27] and specifies a Distributed Agent Environment (DAE) composed of the following entities: Region, Place, Agency and Agent. An Agency provides the runtime environment for Agents. A Place provides a logical grouping of functionality inside an Agency. Finally, a Region is using a Region Registry to keep information about all the Agencies, Places and Agents that reside on the corresponding Region. Each Agency provides several services that are usually common to all agent-based development platforms (e.g., Execution, Communication, Management, Persistence, Registration, Security and Transport). Communication could be performed via CORBA IIOP, Java RMI or plain socket connections while the last two are supported by the Secure Socket Layer (SSL).

In our implementation each network management entity (i.e., NSA, RNCA, and NBA) is created in a separate Agency. The NA and PA reside on the same Agency. The creation and initialization of these agents are described in more details in the following sections. In Figure 7, we show an Agency that contains a NBA that serves three active mobile users (represented by

the respective TA, MUA and VHEA). TA is also hosted in the same Agency with NBA while the case of a mobile handset is examined. Each user invokes one service (represented by a Service Agent). Examples of service appearance on the device of a mobile user are shown in Figure 8. Figure 8(a) shows a service for meteorological information and Figure 8(b) shows a service that provides a map of the surrounding area where the user currently is. The mobile handset runs J2ME and allows the VHE's framework to communicate with the mobile agent based framework.

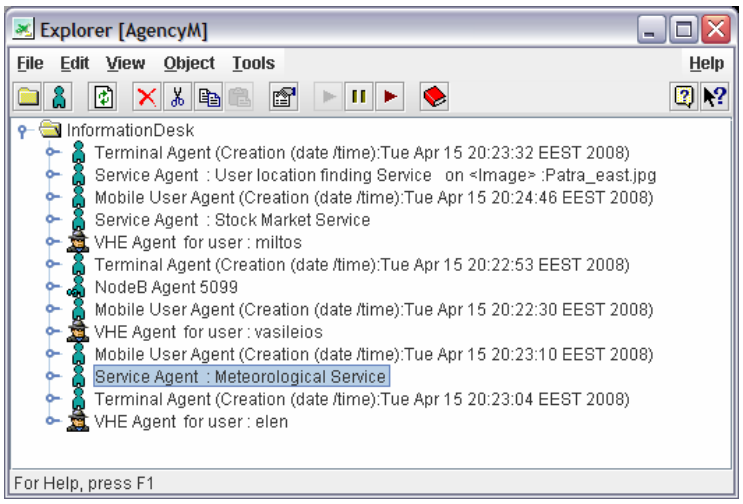

Figure 7: An agency with a NBA, and three active mobile users.

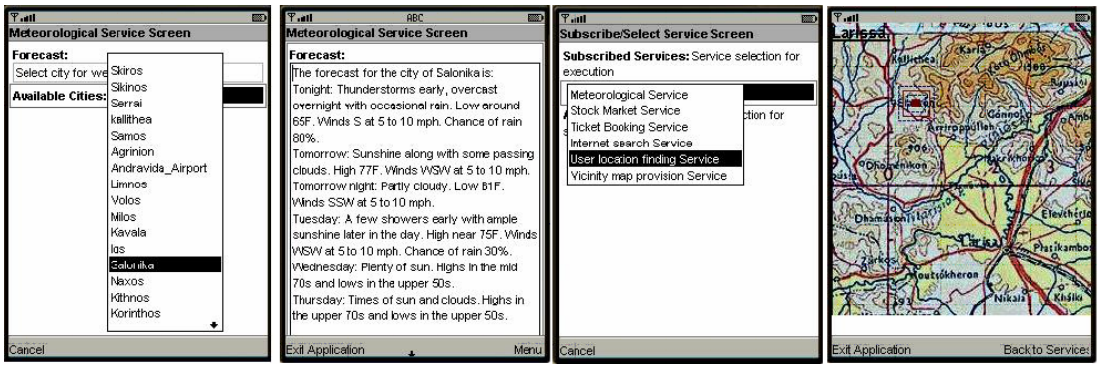

Figure 8: (a) meteorological information and (b) map service - in a mobile phone.

In Table 2, we provide an indicate class structure for a mobile and a stationary agent. Grasshopper provides several agent-specific methods that are invoked by the hosting Agency prior to the agent's action. These methods may be overridden by the agent for certain tasks (e.g., to release resources and references to other objects before migration, cloning or removal). Such methods are: *Copy(·)*, *move(·)*, *clone(·)* and *remove(·)* while there exist methods for *before* and *after* the aforementioned actions (e.g., *beforeCopy(·)*, *afterCopy(·)*). Upon agent creation the hosting Agency automatically invokes the agent's *init(·)* method for initialization. An important method is the *live(·)* method that implements the agent's autonomous behaviour. *live(·)* is called when an agent migrates to a new Agency/Place.

Table 2: Stationary and Mobile Agent

Stationary agent	Mobile agent

public class RNCAgent extends
de.ikv.grasshopper.agent.StationaryAgent implements IRNCAgent{
*//** Upon its creation, sends "active" message to NA **/*
public void init(Object[] creationArguments) {}
*// ** Invoked by the Grasshopper GUI**//*
public void action() {}
*//**Searches for active agents. In case of a RNC failure may find active Node B agents already registered to this RNC**/*
public void live(){}
public void beforeRemove() {}
public String getAgentName() {}
public String getDescription() {}
*/** Send heart beat messages to the Network agent */*
public void setMaintainanceMessage() {}
*/** Get reference to NA's interface in order to be able to communicate with NA**/*
public INetworkAgent getReferenceToNetworkAgent() { }
*/** Get reference to Provider Agent **/*
public IProviderAgent getReferenceToProviderAgent() {}
*/** Authenticate a user to the Network.**/*
public boolean authenticateUser(TpAuthenticationInfo userAuthenticationInfo) {}
*/** Provide to the requesting entity additional information concerning a service (e.g servicing network, required service parameters etc). .**/*
public TpService getAdditionalServiceParameters(TpServiceInfo ServiceInfo) {}
*/** Subscribe a user to a service.**/*
public boolean subscribeToService(TpServiceSubscriprionInfo serviceToSubscribe) {}
*/** Provide the available services from the current network.**/*
public TpServices[] getCurrentNetworkServices() {}
*/** Find active NBAs that exist in a region. It is called when the NA is created. * */*
public boolean findActiveNBAs() {}
*/** Delete an NBA that is not active for a period of time*/*
public void deleteNBA(Maintainance NBAInfo) {}
*/** Receive Maintenance messages from the monitored network entities (i.e active NBAs) .**/*
public void setMaintainanceMessages(Maintainance NBAInfo) {}
*/** Create a multicast group for NodeB agents in order for them to invoke their Maintenance procedure (i.e to send heart beat messages to RNC agent). */*
public void invokeMaintainanceProcedure() {}
*/** Maintenance procedure pertains to the monitoring of state of the NodeB that are registered to the current RNC agent This procedure is invoked in regular intervals and is implemented in a separate thread in each RNC agent. **/*
private class MaintainanceDaemonRNC implements Runnable {}
}
	public class MobileUserAgent extends MobileAgent implements IMobileUserAgent {
	public void init(Object[] creationArguments) {}
	*// ** Invoked by the Grasshopper GUI**//*
	public void action() {}
	*/** Restore user GUI and state of the agent **//.*
	public void live() {}
	public void beforeMove() {}
	public void beforeRemove() {}
	public void afterCopy() {}
	public void beforeCopy() {}
	public void afterMove() {}
	*//** Get the name of the Agent**//*
	public String getAgentName() {}
	*//** Get Agent's description **//*
	public String getDescription() {}
	*/** Get a reference to the nearby NBA's interface in order to be able to communicate with NodeB Agent * */*
	public INodeBAgent getReferenceToNBA() {}
	*/** Get a reference to the servicing VHE agent* */*
	public IVHEAgent getReferenceToVHE() {}
	
	*//**Methods accessed through IMobileUser interface*/*
	*/** Authenticate the user to the Network.*
	public boolean authenticateUser(TpAuthenticationInfo userAuthenticationInfo) { }
	*/** Start a service that has been requested by the user * */*
	public boolean startService(TpServiceInfo serviceInfo) {}
	
	*/** Subscribe the user to a new service.* */*
	public boolean subscribeToService(TpServiceSubscriprionInfo serviceInfo) {}
	
	*/**Set authentication results of the user to MUA. * */*
	public boolean setAuthenticationResults(results){}
	/** The results of user authentication are passed to MUA after the latter has migrated to another agency-place.
	public boolean setAuthenticationResultsAfterMove(results){}
	
	*/** Retrieves the current location of the MUA * */*
	public GrasshopperAddress getCurrentLocation() {}
	}

We note two characteristics of Grasshopper:

- Agent Communication is performed (either synchronously or asynchronously) with the aid of proxies implemented as interfaces on the called agent. Proxy objects are easily found through an agent directory service provided by the platform. These proxies are easily developed and can communicate with non agent applications. However, the method signature on the called agent should be known a-priori. Other agent platforms use ACL (Agent Communication Language) messages, which are asynchronously delivered to communicating entities.

- Mobility in MA platforms is categorised as: strong and weak. Strong mobility refers to the process that enables an agent to move, transparently, as a whole by retaining its execution state (e.g., instruction pointer) across migration and resume its execution on the new host right after the instruction that triggered the migration. Weak mobility enables the transfer of application code to another host, and at destination, the code may be run into a newly created executing unit or it may be linked into an already running one. Grasshopper supports the latter scheme.

The above features of Grasshopper influence any multi-agent system that is build on top of it and the development of MAs. Synchronous agent message exchange accounts for a relatively fast communication service and agent migration type pertains to the programming of an agent (e.g., precisely defining into the agent code where, when, what and how the agent should do).

4.2 Platform description and service scenarios

The description of the proposed architecture is divided in three main sections:

- Setup /Initialisation of the agent platform: describes how the network management agents are created and how they establish a connection with each other. We provide a detailed presentation of how the entire MA platform is activated (creation sequence of agents and databases) and how failures are handled.
- User interactions with the platform: describe the possible VHE requests a user may submit to the platform and how these requests are handled.
- Service Provision Scenarios: are categorised according to the location of service execution: (a) Service execution into the Visited Network, (b) Service execution into the Home Network or Service Provider and (c) Service execution in the user's device.

The proposed VHE platform allows users to request services from different devices with various capabilities (fixed PCs, cellular phones) and, thus, different approaches have been designed for the initiation and presentation of the platform to the end-user. If a user wishes to access the service through a fixed PC or a laptop, service is presented through a simple java-enabled web page. The user is authenticated by the platform. Subsequently, an agent server starts running in his/her terminal to accommodate the agents that are responsible for service management and provisioning. In the mobile handset case, terminal capabilities are also taken into account.

4.2.1 Setup/Initialization of the agent platform

In the setup and initialisation of the agent platform a NSA is created for handling networks and services registration. Subsequently, a NA is created and the NSA is informed accordingly. NSA provides to the NA a list of the available networks, if any, their addresses and the status of all NAs that have registered to the NSA with the same procedure. Such operation is shown in Figure 9. Subsequently, the NA creates the PA that is responsible for the services offered in the network and also maintains information for the services offered in other networks registered in the same NSA. During network initialisation, the PA communicates with the NSA, in order to receive a list of the services offered by other networks. In case of network failure the latter asks the NSA if there is any change in the list that has been received since the initialisation of the network. Consequently, after the creation of the NA and PA, the RNC agents are created (one RNC agent for each RNC of the network). Each RNC agent, upon creation, advertises its existence to the NA. Such "presence" messages contain configuration parameters indicating the status of each node. Subsequently, in each RNC, the NBAs are created. Each NBA, upon creation, advertises its existence to its controlling RNC, by sending a presence message containing configuration parameters for the status of each node. The hierarchical creation of RNCA from NA, and that of NBA from the respective RNCA is similar.

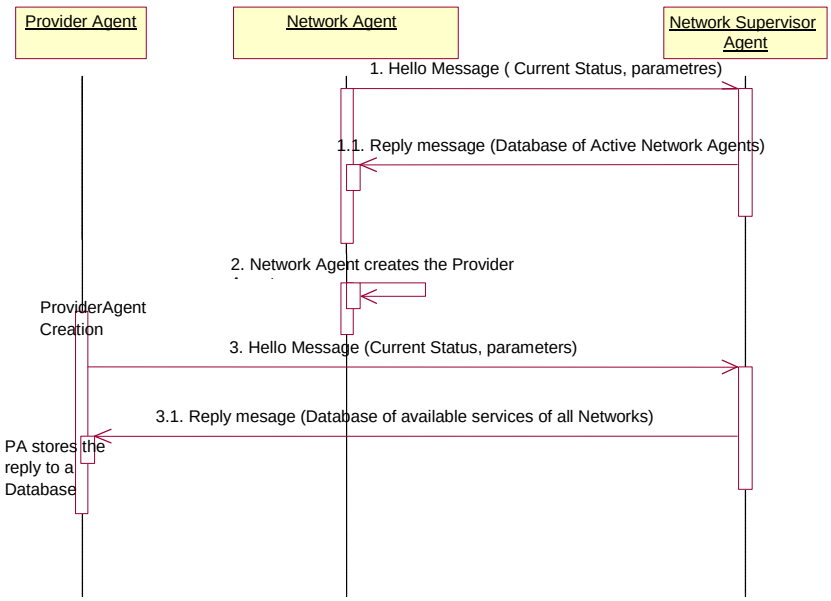

Figure 9: Network initialization

The NA periodically requests and receives "heart-beat" messages from the RNCAs of the network. These messages are used for monitoring the operation of RNCAs. When a RNCA receives a request for a "heart-beat" message, it replies with a message that contains information about its status and current users. If a NA does not receive a "heart-beat" reply from a RNCA, within a certain time interval, then this RNCA is considered non-operational. If a RNCA is set to "failed" status then the NA does not forward any traffic to it. In the occurrence of traffic forwarding from a failed RNCA to the NA, the latter changes the status of the previously "failed" RNCA to "active" and also sends a "heart-beat" message to this RNCA to reply with a status message. The same maintenance procedure exists between each RNCA and their supported NBAs, where each RNCA requests periodically maintenance messages from its supported NBAs.

The NA maintains a database that contains information about all RNCAs (e.g., network address, user service statistics and status). A similar database is also maintained in each RNCA with information about the adjacent RNCAs. It is periodically updated by the NA. Finally each NBA maintains a database with information about the adjacent NBAs that is periodically updated by the RNCA.

Moreover, the NA maintains a database of the existing active networks. In this database, information concerning the addresses of the NA and their functional status is stored. Such information is updated at regular time intervals by sending message reports from the NSA. Moreover, two additional databases are defined in each PA. These databases contain information about the network-wide available services and platform available services respectively. For each database, frequent updating processes are followed.

4.2.2 User interactions with the platform.

In the following paragraphs the possible user interactions with the agent platform are described. We also elaborate on the platform functionality required for serving user requests.

4.2.3 User Registration to the Network

The user registers to the network either by providing his/her credentials through a web interface of the mobile agent platform or by using a fixed PC or a portable/handheld device.

4.2.3.1 Mobile device Case (cellular phone)

Assuming a mobile user opens his/hers device and enters a Personal Identification Number. After the registration and authorization of the user a MUA is automatically created to represent the user in the fixed network. Since the mobile device does not have the capability to accommodate this agent, the latter is located at the serving Node B. At the same time, the NA initiates a process to find the service profile of this user. If the user is logged in his/her HN, the NA simply requests the service profile from the PA. In case of a roaming user, the NA requests the service profile from the corresponding PA of the HN. In the latter case, the NA transfers the user's service profile from HN PA to the visited network PA. The PA, after receiving the roaming user's service profile, compares the user's subscribed services (taking also into account his/hers preferences) with the services being available to the current network. If there is (are) subscribed service(s) included in the user's service profile that can be offered from the current network then the user is informed on this option. Thus, operational cost could be significantly reduced since the service(s) will be offered through the current network.

Subsequently, the PA creates the VHEA which contains the user's profile and sends it to Node B that currently serves the user. The VHEA has information on the services that are available in the visited network. The MUA receives from the VHEA the user's service profile and according to his/her terminal capabilities, presents the subscribed services to the mobile user. The capabilities of the mobile device are expressed with CC/PP. These capabilities have been forwarded from the TA to the MUA.

The user service profile that is transferred from his/her HN is not the same with the profile that the MUA finally receives from the VHEA. The user service profile received from NA contains information about the subscribed services, the provider for each service, service invocation parameters, user preferences/customisations and configuration parameters that should be applied to the visited network in order to provide subscribed services. In this phase, VHEA contains a limited part of this profile and, thus, has only the list of subscribed services. This is due to the fact that the user may choose not to use any of these services. Hence, pointless data transfer is avoided. When the user requests a service, the appropriate service information is obtained by the VHEA from the PA of the visited network that maintains the complete user's service profile. This process is depicted in Figure 10.

In case, the user is a subscriber of the current network the procedure discussed in the previous paragraph is simpler. The NA simply informs the PA on the ID of the user. The PA then finds the user's service profile, creates the VHEA, and dispatches it to the current location of the MUA.

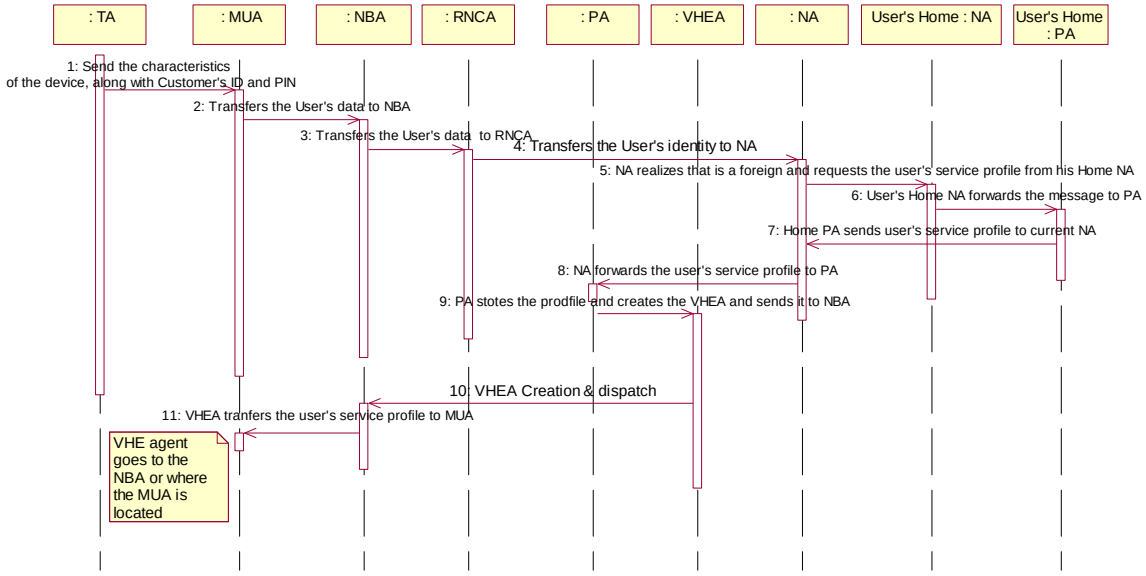

Figure 10: User registration – cellular phone

4.2.3.2 PC, laptop or Personal Digital Assistant (PDA) Case

A quite similar scenario is adopted for devices such as fixed PC, laptop and PDA. The same procedures are followed for VHE service registration. For a fixed PC the communication between the TA or VHEA and network agents could be regarded as direct, since Node Bs and RNCA are absent. When a PDA or laptop is used, the communication is achieved through Access Points (AP). The message exchange for user's service profile discovery is similar to that described in the previous paragraph. The service profile is transferred from his/her HN to the user's device. Hence, the VHEA transfers the complete service profile to MUA. The latter presents the services to the user, by taking into account his/her preferences.

4.2.4 Service provision

Service provision is executed after the registration of the user to the agent platform. The network status and the location of each agent for all possible devices is shown in Figure 11

The user requests a service through the MUA. The latter relays this request to the VHEA. VHEA, which is aware of service execution details and user's service profile, proceeds with the invocation of this service. When the VHEA contains a limited version of user's service profile, it requests the appropriate service parameters from the corresponding PA. VHEA obtains the additional information needed for the service invocation, service parameters and service providing network entity and sends a service invocation request to the service provider entity. This request is transferred through the following networks entities: VHE Agent→ Node B Agent → RNC→ Agent Network Agent → Service Provider, depending on user's device and access network (Home or Visited). In the case of a fixed PC, PDA and laptop, Node B, RNC and Network entities are not participating in this communication.

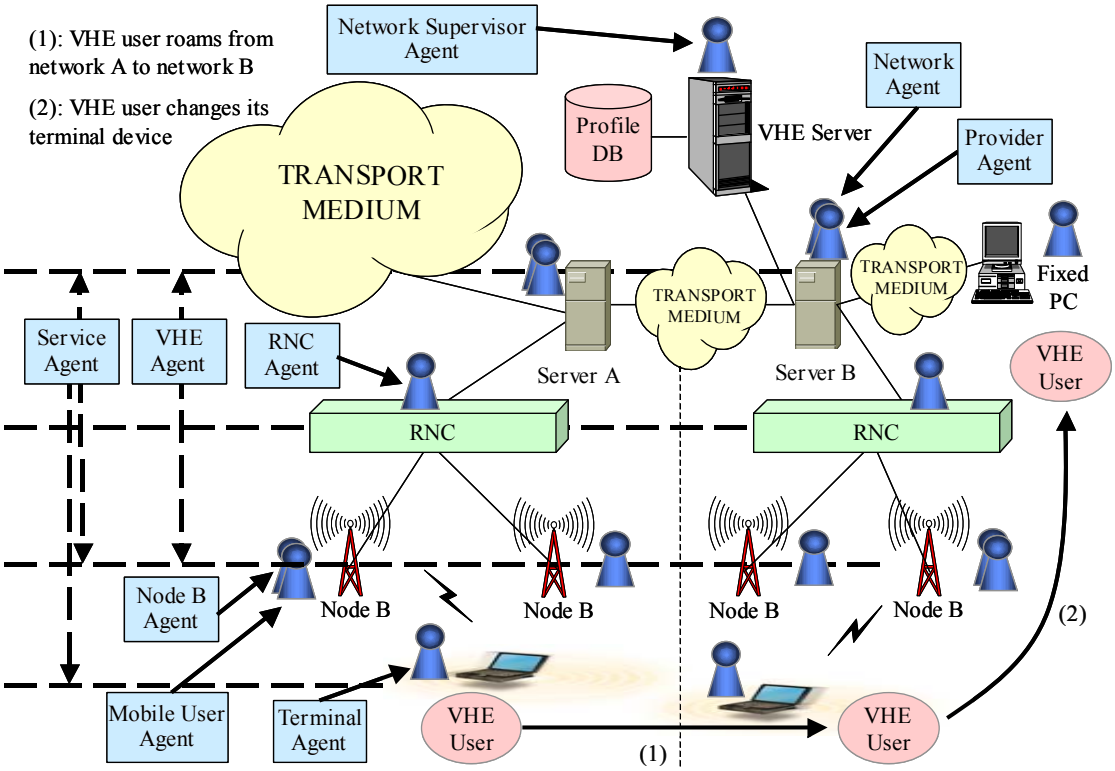

Figure 11: Agent-Based VHE Architecture

Service provision scenarios are categorised according to the node where service execution took place:

Service execution into the Visited Network: According to this scenario the service execution is performed into the visited network. Service data and logic must be transferred to the visited network. Hence, when the Service Provider or HN receives a service request, a SA is created containing both service data and logic. The SA is dispatched to the NA that requested the service. The latter either executes the service locally and forwards the results to the MUA or forwards the SA to a Node B (e.g., mobile handset) for executing the service there and passing the results to the MUA.

Service execution into the Home Network or Service Provider: This scenario assumes that service execution is performed either in the HN or in the Service Provider. The results of the service should be conveyed to the visited network. The service request and provision process is similar to the one previously discussed with the exception that the SA transfers service results. The migration of SA to the visited network is independent of the final location of service execution. Finally, when the user stops the service, a "stop" message is forwarded to VHEA from MUA. Such message propagates to the entity that provides that service, (e.g., a network or service provider). Network or Service provider replies with a positive acknowledgment.

Service execution in the user's device: This scenario assumes service execution in the device of the user as long as the local TA has informed the network on its capabilities. Since the mobile device is capable of accommodating the execution of agents, a VHEA is created from the PA of the serving network and is dispatched to the user's device. The user requests a service. The MUA delivers this request to the VHEA. The latter after receiving further service information from the PA, sends a service invocation to the entity that provides this service (e.g., user's HN or Service provider). The PA of the Service provider creates a SA. SA dispatches to the NA of the current network. The NA forwards the SA to the NBA that the user currently is. The latter forwards the SA to user's device. The SA executes and delivers the service results to the MUA that handles the presentation of the service.

5. PERFORMANCE EVALUATION MODEL

In this section we describe the simulation model adopted for assessing the performance of the proposed MA framework. Two cases have been considered and compared: (a) Mobile Agent (MA) service provision and (b) Conventional Service Provision (Without Mobile Agents, indicated as WMA). Each user issues a number of service requests irrespectively of his/her current location. The simulation study consists of two parts. In the first part the services offered to roaming users are monitored for a single day, whereas in the second part the capacity of the network entities are studied for half hour.

In both parts of simulation the user and system behaviour is the same. The user moves between Node Bs and RNCs under the same or different networks with specific probabilities. Considering the hierarchical structure of the simulated model and assuming that users handoff from one Node B to another, we assigned control probabilities. Furthermore, since roaming users exhibit different moving behaviour, three user classes have been defined to model this behaviour. These classes are specified according to speed levels since handover occurs faster to fast moving users than slow moving users. Finally, different service types have been defined according to their activation probabilities and bandwidth requirements.

5.1 User movement

In this paragraph we discuss mobility model assumed in our study.

RNC Control Probabilities

$P_{RNC}(i,j)$ denotes the probability of a user moving from RNC_i to RNC_j (see Figure 12). The user remains in the same RNC_i with a $P_{RNC}(i,i)=0.6$ or switches from RNC_i to RNC_j with $P_{RNC}(i,j)=1- P_{RNC}(i,i)$.

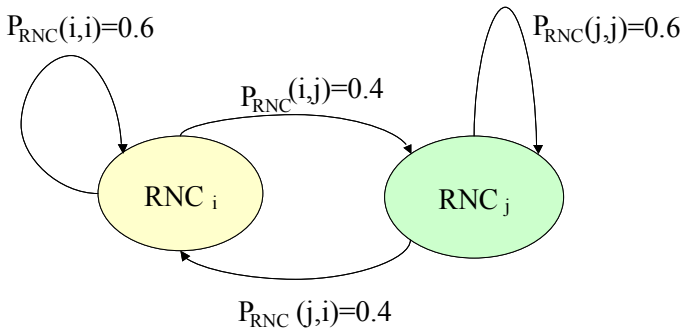

Figure 12: RNC Transition Probabilities

Network Control Probabilities

$P_N(i,j)$ denotes the probability of a user moving from network i to network j. A user remains in his/her home network with $P_N(i,i)=0.7$ and roams to another visited network with $P_N(i,j)=1-P_N(i,i)$ after a specific time period elapses (i.e., handover time). A user in a visited network, the probabilities are $P_N(i,j)= 0.3$ to return to his/her home network, $P_N(i,i)= 0.6$ to remain at the same network and $P_N(i,z)= 0.1$ to move to another foreign network (Figure 13).

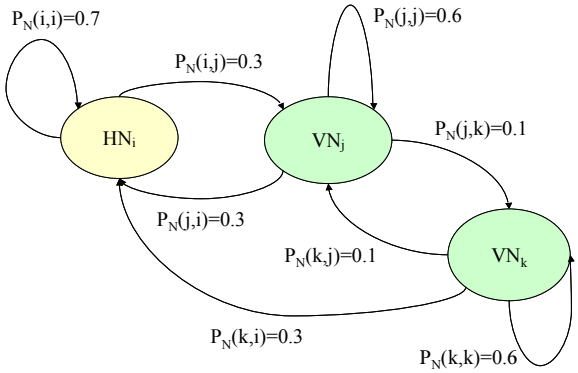

Figure 13: Network Transition Probabilities

5.2 User arrival/departure process

Users enter the system according to a Poisson process. New users are uniformly distributed to the network. Assignment to home or visited network is equally probable. Users leave the system at random

5.3 Time parameters

Each user movement is associated with some corresponding signalling within the interconnected networks in order to be continually and transparently served. Such signalling imposes a certain time cost. We define the following random variables:

- T_1: time required for invoking a service when the user resides within his/her HN.
- T_2: time required for a service invocation when the user resides in a VN.
- T_3: time required for user roaming from Node B_i to Node B_j in the same network.
- T_4: time required for user roaming from Node B_i to Node B_j in different networks.
- T_5: time required for user relocation from RNC_i to RNC_j in the same network.

Finally, we model the RNC handover time as a constant value a=1.5 sec. Therefore, we assume $T_4=T_3+a$. The above random variables follow the Gaussian distribution (parameters shown in Table 3).

Table 3: Service and Network Timers

Time Costs		Value (sec)
Service window time	Constant	360
Service downloading from the same network (T_1)	Mean	0.2
	Standard deviation	0.05
Service downloading from different network (T_2)	Mean	183
	Standard deviation	5
User relocation from Node B to Node B (same network) (T_3)	Mean	2.012
	Standard deviation	0.331
User relocation from Node B to Node B (different network) (T_4)	Mean	183
	Standard deviation	5
User relocation from RNC to RNC (T_5)	Constant	1.5

A very important aspect in our simulation is the time taken for a user to cross each cell (i.e., the Cell Residence Time (CRT)). We implemented the Generalized Gamma Distribution

(GGD), to simulate the time that a user remains in a cell, which is considered the best fit for cell residence times [22].

We adopted the Pareto distribution for modelling of User Think Time (UTT) and User Service Request Time (USRT). UTT is the time a user interacts with a service. USRT is the time interval between two user service invocations. Finally, *Service Retrieval Time (SRT)* is defined as the time for a service being downloaded from a service provider. We use the time costs measured from Table 3 to generate Gaussian variates. If a user requests a new service within a predefined window time (service cache time) then this service is not retrieved from the respective service provider (i.e., a Service Retrieval Time (SRT) is not added).

5.4 User classes

In our simulations we considered three classes of roaming users: High, Medium and Low Mobility Users (HM, MM and LM) to model the user moving behaviour. New users are generated with probabilities P(HM)=0.2, P(MM)=0.3 and P(LM)=0.5. Each user class is characterised with different GGD parameters.

5.5 Service types

Three types of services are offered to users. These services are activated stochastically according to their respective activation probability and require different network resources (Table 4). Service type A represents a voice call, service type B corresponds to service applications requiring high bandwidth and, finally, service type C is a WWW browsing service.

Table 4: Service types - activation probabilities and bandwidth requirements

Service	Activation probability	Bandwidth (Kbps)
A	0.5	13
B	0.3	256
C	0.2	128

The user service invocation has been modelled according to user's current position. When a user is on a VN he/she requests a service with probability $P(S_i)$= 0.6 from his/her HN, 0.3 from his/her Current Network (CN) and 0.1 from Another Network (AN) (Figure 14–left). When a user is on his/her HN requests a service to be executed from his/her HN with $P(S_i)$= 0.6 and 0.4 by AN (Figure 14–right).

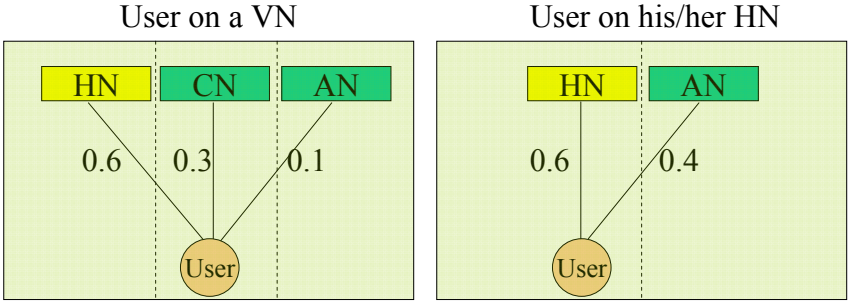

Figure 14: Service Request probabilities.

5.6 Network and Line Capacity

The capacity of each network entity has been modelled according to [24]. Table 5 shows the network components used in our simulation study along with their respective capacities C_{Net}, C_{RNC} and $C_{Node\ B}$. The capacity of each network is obtained by multiplying the number of its RNCs with their capacities, while the capacity of a RNC is obtained by the number of its supported Node Bs times the capacity of each Node B. Equation (3) summarises the above relation of capacities:

$$C_{NET\,i} = \sum_{j,k} \left(C_{RNC}\right)_j + \left(C_{NodeB}\right)_{j,k} \quad (3)$$

where i indicates that network i has j RNCs and each RNC has k Node Bs attached to it.

Table 5: Network components and their respective capacities

Entity	Capacity (Mbps)	Number
Network (Home or Visited)	1830	5
RNC	366	5 per Network
Node B	18,3	20 per RNC

Furthermore, the line capacity is modelled that connects the aforementioned network entities. Line Capacity 1 (LC_1) is the capacity of the line that connects a Node B to a RNC which is equal to 1830 Mbps, whereas LC_2 refers to the capacity of the line that connects a RNC to the Network which is equal to 366 Mbps. The line capacity that connects a Network to the other Networks is equal to 1830 Mbps divided by the total number of Networks in the simulation. Finally, the total line capacity of the simulated network is the LC_3 and is equal to 1830 Mbps. The line capacities are related with the following equations:

$$LC_3 = \sum_{j=2}^{L} \left(LC_2\right)_j \quad (4)$$

$$LC_{2\,j} = \sum_{k=2}^{R} \left(LC_1\right)_k \quad (5)$$

where j represents the j^{th} RNC with k Node Bs attached to it.

5.7 Simulation Scenarios

We simulated and compared the performance of our framework against a conventional service provisioning scheme. In the subsequent paragraphs we analyse each respective simulation scenario and the relevant assumptions.

5.7.1 Scenario with Mobile Agents (MA)

The user enters into the coverage area of a Node B that belongs to his/her HN and immediately requests the execution of a service. Service Request Time is Pareto distributed. Service Retrieval Time is normally distributed. User Think Time is also Pareto distributed. If the time of the next service request is less than the time window then the service is considered stored to a local cache. In the first service request the local cache is considered empty whereas in subsequent invocations the service (i.e., the Service Agent) is either fetched from his/her HN (serving as Service provider) or retrieved from the local cache. These user interactions are occurring until a handover takes place. Handover Time is Gamma distributed. The user after the handover, starts again all the interactions described in this paragraph until the next handover. A pictorial representation of the above assumptions-conditions is provided in Figure 15.

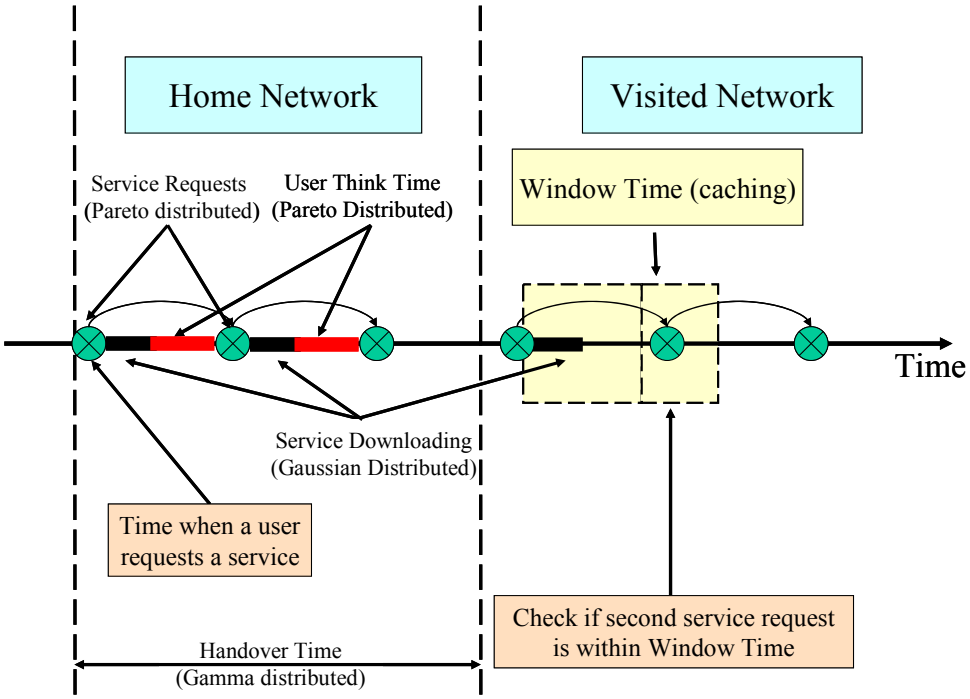

Figure 15: Platform simulation assumptions

5.7.2 Scenario Without Mobile Agents (WMA)

In our simulations we consider a conventional service provision setup that realises the VHE concept without MAs (labelled as Without Mobile Agents – "WMA"). Users' requests are forwarded from the visited networks to the users' home network and service is fetched from users' home network to the network where is currently resides. In the absence of MAs, each time a user roams to a new network and requests a service, the service is retrieved from the user's home network and is available during the user's current session.
In WMA we make the following assumptions:

- The same user mobility model is adopted (i.e., same control probabilities).
- Time costs are identical to the previous case (mobile agents) but are charged differently:
  1. Time cost for a Service Request Time, that is performed in the Home Network, is in the range of 0,2 sec (short time cost)
  2. Time cost for a Service Request Time that is performed in a Visited Network is in the range of 183 seconds (long time cost).
  3. There is no caching of services in the VNs. In the HN there is no need for caching since time cost for a service invocation is extremely low.

Table 6 summarises the service time costs charged to each service invocation for both aforementioned scenarios

Table 6: Time costs charged to each service invocation by users

Service / Network		MA	WMA
Service reuse < Service window	Home Network	Short	Short
	Visited Network	Short	Long
Service reuse > Service window	Home Network	Short	Short
	Visited Network	Long	Long

6. PERFORMANCE EVALUATION RESULTS

The infrastructure of the system we used to measure our platform is shown in Figure 16. The testing system is a LAN with three workstations, two PCs and a laptop, with the last connected to the Internet through a T1 line. A peer workstation located in the USA is used for accommodating roaming users (visited network). The measurements obtained from this system are used as input parameters in the simulation of the MA and WMA cases.

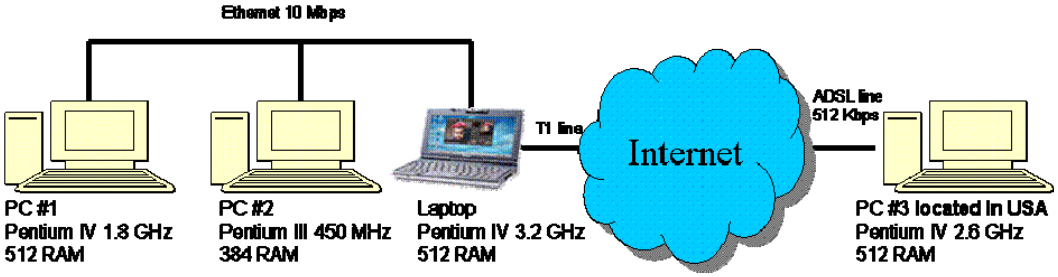

Figure 16: Performance Model Simulation

6.1 System Metrics

Although, the simulation study performed based on the aforementioned user scenarios, we distinguished user's movement (handovers) in the HN from the ones in the VNs to present the different behaviours of both systems in HN and VNs respectively.

In equation (6), we define the Service per Cell (SpC) metric as the ratio of cell residence time to the number of Requested Services (RSi) experienced before each handover occurrence (i):

$$SpC_i = \frac{CRT_i}{RS_i} \qquad (6)$$

In Figure 17 (enriched with second order polynomial fits), are depicted the handovers occurring in the two simulated systems. Handovers indexed from 1 to 74 occurred at the HN where both systems exhibited quite similar behaviour (i.e., approximately the same number of services are invoked) whereas, handovers indexed from 75 to 138 occurred in VNs where mobile agents system enable the user to invoke more services before the handover than in the conventional system.

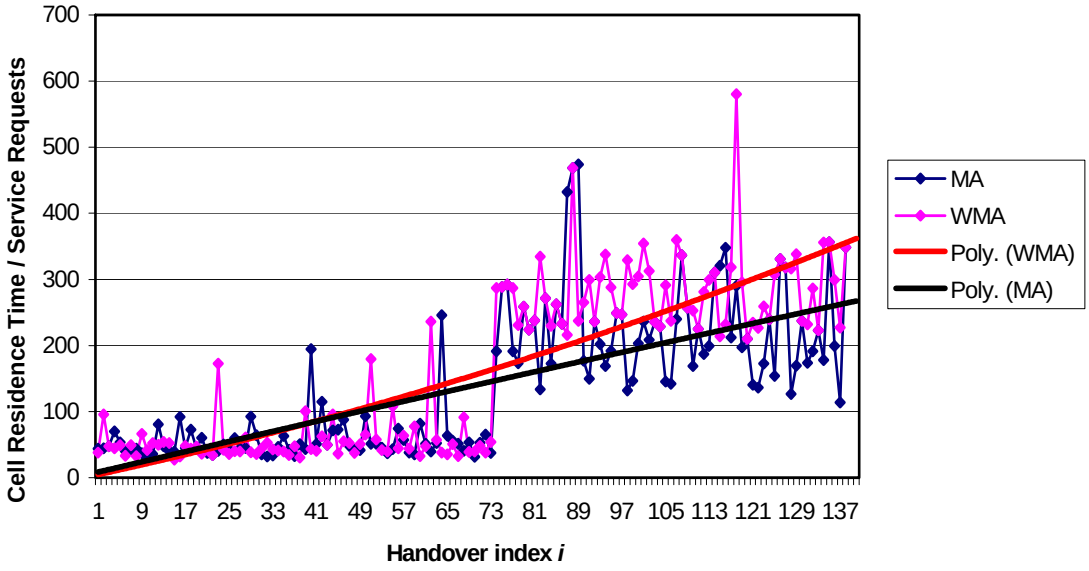

Figure 17: Cell Residence Time/Service Requests vs. Handovers

In equation (7), the Service Usage Time (SUT) is defined as the aggregate time for services requested and used by the user before handover occurrence (i):

$$SUT_i = \frac{1}{CRT_i} \sum_{j=1}^{RS_i} (SRT + UTT)_j \qquad (7)$$

where j indicates the number of the requested services before handover (i).

In Figure 18 (enriched with second order polynomial fits), we plot the Total Service Time versus the number of handovers performed during the simulation. Similarly to the previous metric, handovers indexed from 1 to 74 occurred in the HN where both systems exhibit quite similar behaviour. Handovers indexed from 75 to 138 occurred in VNs. In the VN, mobile agents outperformed the conventional system as they achieved an improved service usage time (e.g., more services provided to the user).

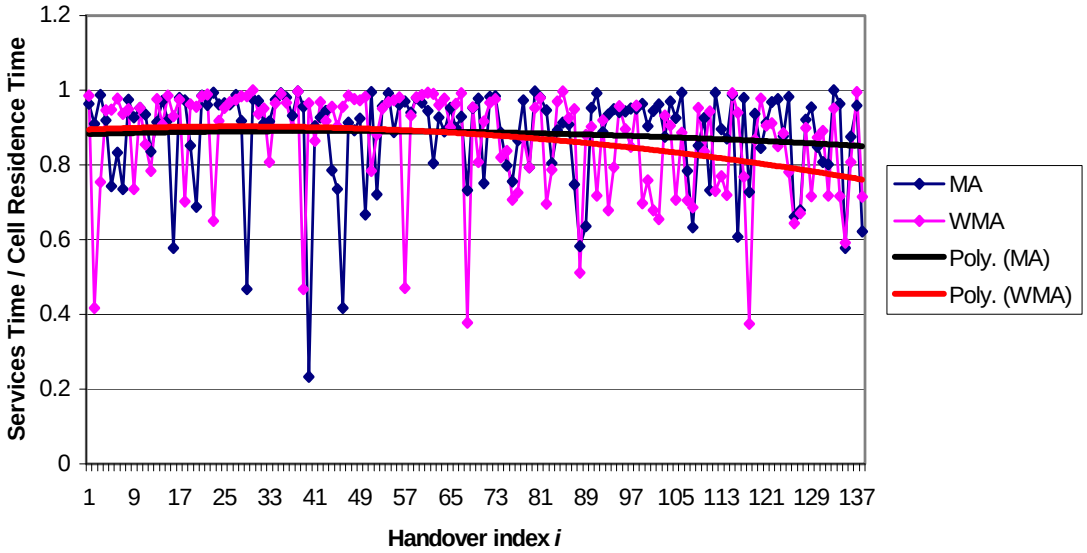

Figure 18: Service Time/Cell Residence Time vs. Handovers

6.2 Capacity

In Figure 19 we plot the Network entities load over time and the node Bs load over time (still with polynomial fits). The simulation is initialized with the random generation of users being placed randomly to Node Bs in the simulated network. The results of the two cases, MA and WMA, are presented and evaluated. Initially, both systems present a similar network load since users that request a service from their home or another network remain for a period of time for the service being downloaded from that network (time T_2 in Table 3). As the time increases users and service requests also increase. It is observed that the load on the WMA increases with a higher rate than the MA system.

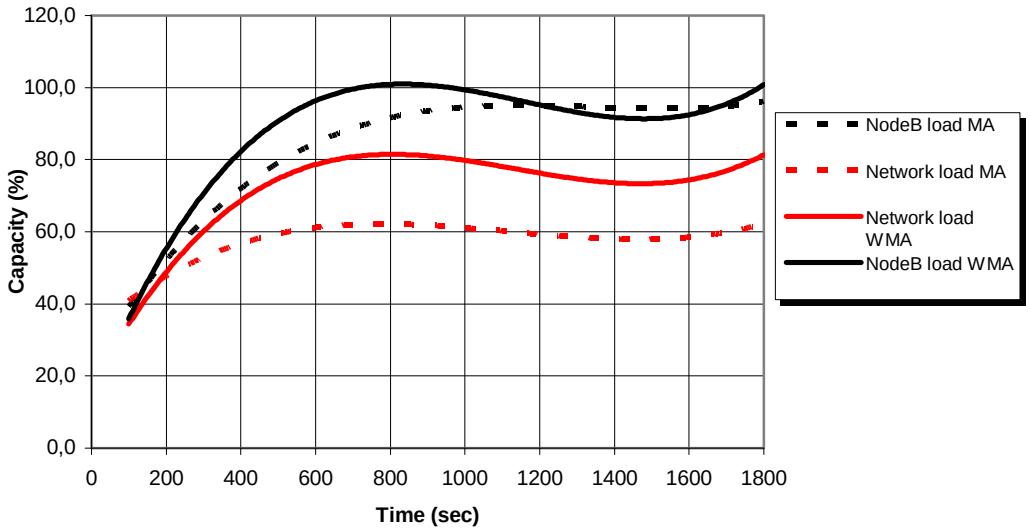

Figure 19: Network and NodeB load on both systems

When a requested service is downloaded (average time 180sec), it is cached on the Node B and offered to the user. Service retrieval for a different user in the same Node B for the same service (same service type and from the same network) is performed from Node B's local cache. The services retrieved from the local cache are provided to the users without affecting the load of the rest of the network. If during the service provision a Node B may request further data through the network a load is added to the network. This explains why average Node Bs load on both cases is significantly higher than this on Networks load during the simulation. Around the 800 sec the average utilization in the WMA system for Node Bs load has reached the 100 per cent and Networks load is 80 per cent, whereas in MA system the utilization is 95 and 60 per cent correspondingly.

During our simulation trials although the average number of users created/roamed/deleted on both systems were in the same range, the services offered by the MA system are on average three times more than the services offered by the WMA system (Table 7).

Table 7: Simulation parameters

		MA	WMA
Users	Created	458.375	412.050
	Deleted	68.953	57.504
	Moved/Roamed	396.684	349.762
Services	Total	598.936	173.605

In Figure 20 we plot (using 3 rd order polynomial approximation) the average of the ratios of:

(a) Services found in the local cache of a Node B (noted as HITS) to the total services invoked on this Node B,

(b) Total services offered successfully to users in the simulated networks, and

(c) Rejected service requests to the users.

As the simulation progresses the MA system outperforms the WMA system. Specifically, in the (b) case the MA/WMA ratio is kept almost constant when simulation time passes 600 sec while in the (a) case the same ratio is kept constant when time passes the 800 seconds. Finally, in case (c) MA's system rejected services are almost the half than the WMA's system rejected services through the whole simulation. From Figures 12 and 13 we observe that MA system offers more services than the WMA system even if both systems have approximately

the same average load on NodeBs (time greater than 800sec). Moreover, as the rejected services to users increase on both systems, the total services decrease and both systems reach a stable state regard to the user service requests after the 1200sec till the end of the simulation.

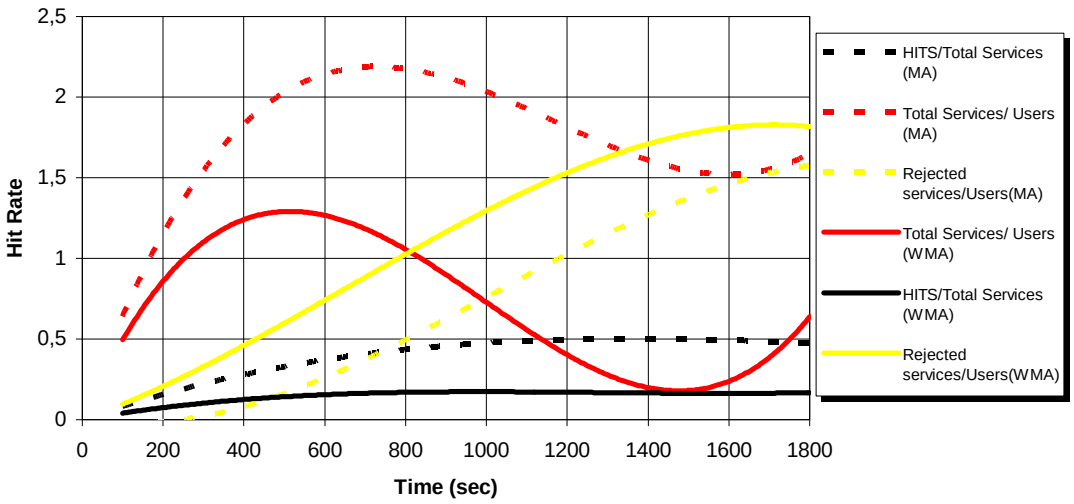

Figure 20: Service provision comparison

7. CONCLUSIONS

The provision of mobile services, nowadays, is a very important aspect in the mobile computing industry. The current technology in the core network and in user terminals is mature enough to support and push the service providers to that direction. VHE service, a concept introduced in 3rd generation networks, enhances the roaming facilities provided to users. The user receives the same service irrespective of the current network, access technology and terminal.

We support the idea that VHE is realizable with the use of mobile agent technology undertaking all the required communication processes in the application level, independently of the underlying network. In this paper, we present such an implementation. Our architecture provides a versatile framework for service provision in telecommunication networks. The use of mobile agents allows for service portability, a very important advantage in heterogeneous infrastructures similar to the ones encountered in the contemporary telecommunications landscape. In our system, service logic is encapsulated in mobile agents thus offering high efficiency, improved security and versatility. A contributing factor to the efficient is the caching of services in the entities of the visited network. Such caching increases efficiency, network utilization and service provisioning capacity. Besides all these aspects, our architecture inherits the obvious benefits stemming from the adoption of Mobile agents (e.g., robustness, autonomous operation).

Our experiments clearly indicated the feasibility of the proposed architecture and its superiority with respect to conventional service implementation/invocation paradigms. Qualitative aspects like platform neutrality were also indicated through the presented performance assessment efforts.

Future planned work includes the description of user profiles and terminal capabilities with ontologies and matching with the service capabilities provided by service providers.

8. REFERENCES

[1] 3GPP Technical Specification 22.121 v5.3.1: "The Virtual Home Environment (Release 5)", June 2002.

[2] 3GPP Technical Specification TS 23.127 V6.0.0 “Virtual Home Environment/Open Service Access” December 2002.

[3] Markus Breugst, Lars Hagen and Tomas Magendanz “Impacts of Mobile Agent Technology on Mobile Communication System Evolution” IEEE Personal Communication Magazine. August 1998.

[4] P.Farjami, C.Gorg F. Bell “Advanced Service Provisioning based on Mobile Agents” Computer Communications, Special Issue: Mobile Software Agents for Telecommunication Applications, Vol. 23, pp. 754-760, April 2000.

[5] Jens Hartmann, Carmelita Georg, Peyman Farjami “ Agent Technology for the UMTS VHE Concept” In Proceeding of the First ACM International Workshop on Wireless Mobile Multimedia in Conjunction with ACM/IEEE MobiCom'98, pp. 203-208, University of North Texas, Dallas, USA, October 1998, ACM.

[6] L. Hagen, J. Mauersberger, C. Weckerle “Mobile agent based service subscription and customisation using the UMTS VHE” Computer Networks, Volume 31, Issue 19, 31 August 1999, Pages 2063-2078.

[7] Tomislav Marenic, Kresimir Mlinaric “Designing Reference Architecture for Providing Virtual Home Environment”, 7th International Conference on Telecommunications (CONTEL), Zagreb, Croatia, June 2003.

[8] IST-1999-10825: VESPER, Website: http://vesper.intranet.gr (last accessed 10/2003)

[9] Lin Songtao, Chen Junliang “An agent-based approach to VHE” proceedings of ICCT2003.

[10] Theo Kanter “An open service architecture for adaptive personal mobile communication” IEEE Personal Communications December 2001.

[11] Songtao Lin, Junliang Chen "Semantic Web Enabled VHE for 3rd Generation Telecommunications", 4th Annual ACIS International Conference on Computer and Information Science (ICIS'05)

[12] J2ME : Java 2 Micro Edition : http://java.sun.com/j2me/

[13] A. Puliafito, S. Riccobene, M. Scarpa, “Which paradigm should I use?: An analytical comparison of the client-server, remote evaluation and mobile agents paradigms”, IEEE Concurrency and Computation: Practice & Experience, vol. 13, pp. 71-94, 2001.

[14] Ravi Jain, Farooq Anjum, Amjad Umar “A Comparison of Mobile Agent and Client-Server Paradigms for Information Retrieval Tasks in Virtual Enterprises”, Academia/Industry Working Conference on Research Challenges (AIWORC'00), Buffalo, New York, pp. 209-214, April 2000.

[15] Robert S. Gray, David Kotz, Ronald A. Peterson, Joyce Barton, Daria Chacon, Peter Gerken, Martin Hofmann, Jeffrey Bradshaw, Maggie R. Breedy, Renia Jeffers, and Niranjan Suri. “Mobile-Agent versus Client/Server Performance: Scalability in an Information-Retrieval Task.” In Proceedings of Mobile Agents, pages 229-243, 2001.

[16] M. Kyriakakos, V. Baousis, S. Hadjiefthymiades, L. Merakos “Ubiquitous Service Provision in Next Generation Mobile Networks”, in the proceedings of the 13th IST Mobile & Wireless Communications Summit, Lyon, France, June 2004, pp. 741-745.

[17] Markus Strasser and Markus Schwehm “A Performance Model for Mobile Agent Systems”, In Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA’97), Las Vegas, Vol. 2, pp. 1132-1140, June 1997.

[18] L. Ismail and D. Hagimont “A Performance Evaluation of the Mobile Agent Paradigm”, ACM SIGPLAN Notices, 34(10), pp. 306-313, October 1999.

[19] M. Baldi and G. Picco “Evaluating the tradeoffs of mobile code design paradigms in network management applications”, In the Proceedings of the 20th International Conference on Software Engineering, Kyoto, Japan, pp. 146 – 155, April, 1998.

[20] G. A. Aderounmu, Performance comparison of remote procedure calling and mobile agent approach to control and data transfer in distributed computing environment, Journal of Network and Computer Applications, v.27 n.2, p.113-129, April 2004.

[21] M. Dikaiakos, G. Samaras, "Performance Evaluation of Mobile Agents: Issues and Approaches." In Performance Engineering. State of the Art and Current Trends, Dumke, Rautenstrauch, Schmietendorf and Scholz (eds). Lecture Notes of Computer Science series - Springer state of the art survey, vol. 2047, May 2001.

[22] M.Zonoozi, and P.Dassanayake, “User Mobility Modeling and Characterization of Mobility Patterns”, IEEE Journal on Selected Areas in Communications, Vol.15, No.7, 1997.

[23] ETSI : European Telecommunications Standards Institute www.etsi.org/

[24] Huawei's End-to-End WCDMA Solution : http://www.huawei.com/publications/

[25] C. Bumer and T. Magedanz, “Grasshopper - an agent platform for mobile agent-based services in fixed and mobile telecommunications environments”, Book Article: Hayzelden, A.L.G.: Software agents for future communication systems, Springer, 1999, pp. 326-357.
[26] European Coordination Action for Agent-Based Computing : http://www.agentlink.org.
[27] MASIF : Mobile Agent System Interoperability Facility: www. omg.org