

Agent Technology for the UMTS VHE Concept

Jens Hartmann¹, Carmelita Görg², Peyman Farjami²

¹ Ericsson Eurolab Deutschland GmbH, Ericsson Allee 1, D-52134 Herzogenrath, Germany
E-mail: Jens.Hartmann@eed.ericsson.se

² RWTH Aachen, Communication Networks, Kopernikusstr.16, D-52074 Aachen, Germany
E-mail: {cg, pemi}@comnets.rwth-aachen.de

Abstract

This paper addresses how the new promising paradigm of Agent Technology can be applied to the provisioning of service ubiquity based on the concepts of the Virtual Home Environment (VHE) being developed for the third generation mobile communication system UMTS. To illustrate the possibilities two sophisticated agent-based telecommunication services, namely the Adaptive Profile Manager and the Virtual Address Book, are described in depth.

Keywords: *Agent Technology, UMTS, VHE, Adaptive Profile Manager, Virtual Address Book.*

1. Introduction

The telecommunication industry is now planning for the evolution towards third generation mobile systems such as the Universal Mobile Telecommunication System (UMTS). UMTS will offer much greater access and interworking flexibility than current networks. Moreover, UMTS will offer much better support for a range of multimedia services and applications for mobile users and end-systems.

It is intended that UMTS will offer users the ability to access their particular set of services from any suitable access point in

either fixed or mobile networks [1]. In addition it has been decided that UMTS will not standardise a large set of teleservices as it was the case for the second generation Global System for Mobile communication (GSM), rather service capabilities will be defined and standardised. Delivery of non-standard services across different network types presents some difficult problems that require innovative solutions. These problems have been identified by the European Telecommunication Standards Institute (ETSI) and the International Telecommunication Union (ITU), and measures have been taken to allow 'service roaming'. The concept providing service portability is called **Virtual Home Environment** (VHE). It is intended that a given service will always appear to be the same from the user's perspective regardless of the access point currently in use.

Agent Technology as the new promising paradigm in networking offers many new opportunities for the efficient provisioning and deployment of telecommunication services. This approach is being pursued in the European ACTS project CAMELEON³ [2].

³ CAMELEON partners: Ericsson Eurolab (D), Mannesmann Mobilfunk (D), Nortel (UK), Vodafone (UK), Gemplus (F), RWTH Aachen (D), IKV (D), CWC (SP), <http://www.comnets.rwth-aachen.de/project/comeleon/comeleon.html>

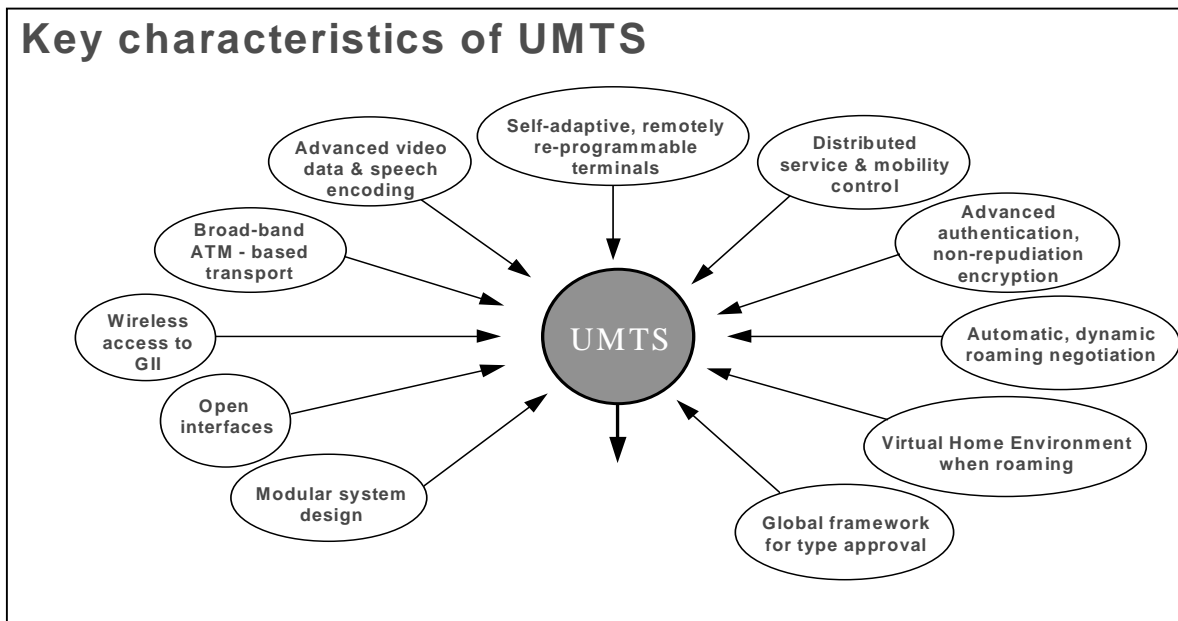


Figure 1: UMTS characteristics

(Source: Neil Lilly, Orange, Asia Pacific Digital Cellular Wireless Communication Conference, 11.-13.09.1996)

2. UMTS

UMTS refers to European standardisation efforts towards a new generation of mobile communication systems where personal communication services will be supported independent of location, terminal equipment, means of transmission (wired or wireless) and underlying network capabilities. UMTS is intended to establish a single integrated system in which users have seamless access to a wide range of new and sophisticated telecommunication services such as high data rate transmission for high-speed Internet/Intranet applications, electronic Multimedia Mail, full-motion video and Electronic Commerce. UMTS will support user bit rates of up to 2 Mbit/s and will provide easy to use, uniform access to services in all environments, e.g. residential, public and office. The key characteristics of UMTS are illustrated in Figure 1.

UMTS users will be able to choose between the service offerings of various networks and service providers, and will have the possibility to invoke them via hand held, portable, vehicular mounted, movable and

fixed terminals in the same way, even when roaming. Service Provider's and Network Operator's will benefit from flexible and easy introduction of new, distinctive telecommunication services based on the VHE (Virtual Home Environment) concept, which is one of the core parts of the system.

3. VHE

The success of entire GSM system [3] relies, among other things, on the possibility to roam between networks – and thus between countries – by using a single subscription. This means that the subscriber is reachable using a single number and receives a single bill from his home service provider. The three most important topics for future mobile users will therefore be the same as they are today for GSM users:

- Easy handling of the desired telecommunication services, including the opportunity to customise the 'look and feel' of services and to subscribe to services 'on-demand'

- Global availability and consistent performance of telecommunication services
- Understandable billing with a single point of contact

However, the future telecommunications world will not be homogeneous, and therefore these goals cannot be achieved easily.

The goal of the VHE concept [4] is to solve these problems by enabling a visited network to obtain information about the user's service provider during the registration procedure and other information such as the user's personalised service profile and the identification of service capabilities needed for the execution of service provider specific services.

Although the physical realisation of a service may differ from one network to another, the VHE concept enables the user to access and to use the service in the same way on any network, see Figure 2. The VHE is currently being standardised in ETSI SMG 1 and ITU SG 2.

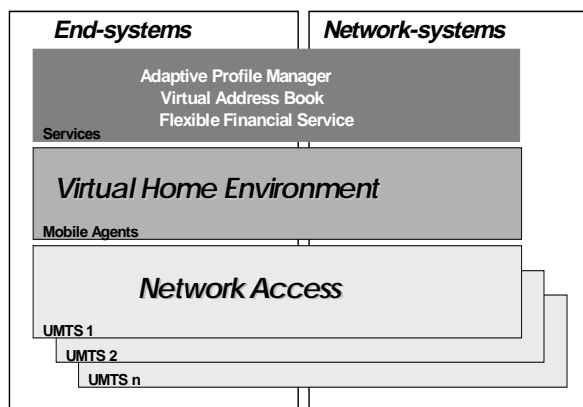


Figure 2: Virtual Home Environment

In summary, the VHE concept supports procedures in a visited environment that provide service portability across network boundaries without changing the service handling the user is accustomed to.

Both the provisioning of additional services by the visited network as well as the emulation of the home environment should be possible within the VHE concept. This flexibility will allow service providers and network operators to compete on the basis of proprietary services and provide great benefit to the user in terms of service choice and service personalisation.

4. Agent Technology

The term “agent” has become a hot topic over the last years in the Information Technology (IT) world as well as in the telecommunication world. However, a commonly agreed definition of what an agent is, cannot be given at this time. We propose the following well-developed definition:

*An agent is a **piece of software**, which is able to perform a specific predefined task **autonomously** (on behalf of a user or an application). An agent is either stationary providing the necessary **intelligence**, or **mobile** so that it can move between **distributed** (possibly incompatible) systems to access remote resources or even meet other agents (or activate them). All agents have capabilities to **co-ordinate**, **communicate** and **co-operate** with the system or with other agents.*

A common approach is to define agents by their attributes, such as autonomy, intelligence, mobility, etc. From a very high level perspective two major types can be identified [5]:

- **Mobile Agents**, in which the mobility of code, data and state is the most fundamental attribute. This allows software entities to roam autonomously through a network and to perform dedicated tasks at specific network nodes, thereby taking advantage of locality;

- **Intelligent Agents**, which are software entities that are able to perform delegated tasks based on internal knowledge and reasoning, where aspects such as inter-agent communication and negotiation are fundamental. Usually mobility is not considered as an issue.

4.1 Agent Languages

The most important programming languages that supporting the agent concepts are **JAVA**, **TCL** (Tool Command Language), **APRIL** (Agent PRocess Internet Language) and **KQML** (Knowledge Query and Manipulation Language).

The platform independence of the **JAVA** language [6] makes it a perfect environment to develop agent-based tools. For example, IBM's Aglet Workbench, currently in alpha release, provides a laboratory for creating JAVA-based mobile agent applications.

TCL is a popular public domain interpreted language developed by Sun. It was originally designed as a language to control various applications and allow them to communicate. TCL is a script language and definitely more user friendly than JAVA. TCL/TK was developed for rapid prototyping and GUI development. Safe-Tcl is one extension for running agents.

April is a programming language developed at Fujitsu Labs and specifically designed for building MAS (Multi Agent Systems) which execute over the Internet. Agents are built as a set of layers on top of the April language. April provides a set of basic features allowing the creation of concurrent processes, TCP/IP based communication between them, high-level communication primitives, list data structures etc.

While the languages mentioned so far are all programming languages, **KQML** is a communication protocol for exchanging information and knowledge. KQML covers both the message format and the message-

handling protocol to support run-time knowledge sharing among agents.

4.2 Agent Platforms

Platforms for service provisioning are continuously evolving. However, the evaluation of the application of Agent Technology for the purpose of providing service ubiquity has to be shown on an existing platform.

Besides providing a system development infrastructure the existing platforms, also called mobile agent systems, support security, inter-agent communication, agent transport protocols, remote messaging, etc. Several platforms are available but most of these products are in their Beta stages. Some of the JAVA-based mobile frameworks available are: (1) *Aglets Workbench* from IBM, (2) *Voyager* from Object Space, (3) *Concordia* from Mitsubishi Electric, (4) *Agent TCL* from Dartmouth College, (5) *Odyssey* from General Magic, (5) *CyberAgent* from FTP Software, (6) *Grasshopper* from IKV, and (7) *JIAC* from the Technical University of Berlin.

A first evaluation with respect to the advantages/disadvantages, stability, availability, portability etc. of these candidates has shown that the **IBM Aglets Workbench** [7] platform offers a promising approach. This Aglets framework consists of a JAVA-based API that enables the creation of mobile agents in an application. Aglets are JAVA objects that have agent mobility. During execution, an Aglet can move to another host, and resume execution there. Like applets, Aglets can move from one host to the next, but, unlike applets, Aglets take their data and state of execution with them.

Aglets possess four main properties: state, behaviour, a unique identity, and location. Aglets gain mobility by using the Agent Transfer Protocol (ATP) and the JAVA Agent Transfer and Communication

Interface (J-ATCI). ATP is an application level protocol for distributed agent based information systems and facilitates migration of Aglets over the network.

J-ATCI is an independent agent protocol enabling agents to move and communicate within a network. J-ATCI reinforces ATP at a higher communication level.

Many Aglets use an itinerary class to specify their travel plan. This class also encapsulates instructions for the Aglets to which hosts to go to and what to do on each host. Aglets are created in an Aglet context. These contexts provide a place for Aglets to communicate with each other. Aglets interact with each other through Aglet proxies, where they send messages to each other. These messages can be asynchronous or synchronous.

Another agent platform is **Object Space Voyager** [8], which is a JAVA-based agent-enhanced Object Request Broker (ORB). It does not use RMI to move agents, instead it has its own object transport mechanism which, as claimed, makes it faster than RMI. Voyager also provides ORB functionality that means remote objects can communicate in a unified environment. Voyager allows JAVA objects to roam a network and continue to execute as they move. And since an agent is just a special kind of object, moving agents and other objects can exchange remote messages using regular JAVA message syntax.

With Voyager a single command automatically enables any JAVA class without altering the class for distributed computing. Voyager can remotely construct and communicate with any JAVA class, even 3rd party libraries without source. Once created, one may move any serialisable object to a new location, even while the object is receiving messages. Messages sent to the old location are automatically forwarded to the new location. Voyager is small. The entire Voyager system is less

than 150K (uncompressed). It is a fully functional agent-enhanced ORB, and does not require any additional software beyond the JAVA Development Kit (JDK) 1.1. Future versions will include a distributed event system, group communications, a distributed directory service, store-and-forward, reliable UDP communications, mobile tracking facilities, and enhanced agent capabilities.

These JAVA based platforms allow to create portable objects and easily distribute them and they are well suited for new developments that need only cooperate with existing software in a limited way. CORBA based agent platforms such as Grasshopper from IKV, have the advantage that systems in other languages can be integrated using the Interface Definition Language (IDL) of CORBA, thus allowing the integration of legacy systems, database systems, and objects and applications written in other languages. Each platform has its advantages and disadvantages and it is the goal of CAMELEON to use both platform types in the prototyping to allow a well founded judgement on the applicability and interoperability of these platforms for ubiquitous tele-communication services of the VHE type.

4.3 Agent Standardisation

Several standardisation groups have been founded to specify and standardise agents and their use under a different focus, including the **Object Management Group (OMG)**, the **Foundation of Physical Intelligent Agents (FIPA)**, and the **Agent Society**.

OMG [9] has submitted a **Mobile Agent Facility (MAF)** specification. This specification focuses on interoperability between the different agent platforms mentioned in the previous chapter. The goal of the MAF is to accelerate the use of

mobile agents by maximising interoperability between mobile agent systems while minimising the impact of the standardised elements on any particular system. Interoperability allows agents to visit more sites because they are not limited by the incompatibility between disparate mobile agent systems.

FIPA [10] is an international non-profit association of companies and organisations that have agreed to share efforts in specifications of generic agent technologies and focus on intelligent agents. Only static intelligent agents are considered at the first stage, whereas agent mobility (although considered as being interesting for some application contexts) will be considered in the future.

The **Agent Society** [11] is an international industry and professional organisation established at the end of 1996 to assist in the widespread development and emergence of Internet related agent technologies and markets. The members of this non-profit consortium consist of leading companies and Institutions in the agent field. In contrast to FIPA the Agent Society is primarily concerned with raising the awareness of Agent Technology by providing opportunities for the industry to meet and discuss issues of common interest, and not with producing the specifications.

To identify the potential of Agent Technology in the telecommunication world a broad range of services has been selected for prototyping showing the possibilities of the UMTS VHE. Out of a long list of possible and extremely promising new services, Adaptive Profile Management, Virtual Address Book, and Flexible Financial Services were chosen to be studied intensively. In the following two sections the first two are described in more detail.

5. Adaptive Profile Management

Current applications in this area provide only simple communication management functionality, for example forwarding all incoming voice calls to a predefined destination such as a voice mailbox or an alternate telephone number. Additionally, within today's mobile environment notifications are supported to inform a user that he has a new fax or email. However, the user has no single means of managing these various communication services, and limited flexibility in specifying when they should be active. For instance, a UMTS user may want to stop all notifications and send incoming voice calls to a mailbox while having a meal in a restaurant. In today's environment he would have to go through a series of different menus on his mobile to accomplish this. In the future a single "soft-key" selection of "DO NOT DISTURB" could automatically accomplish his requirements by triggering his adaptive profile manager to deal with incoming communication requests in a pre-determined way [2].

The Adaptive Profile Manager (APM) will provide the following capabilities:

- Provisioning of personalised call management capabilities for the end-user, i.e. the user can personalise the handling of calls by creating a set of rules (routing/filtering) as part of the user profile which the system uses to intelligently handle incoming communication requests
- Ability to flexibly combine services handling

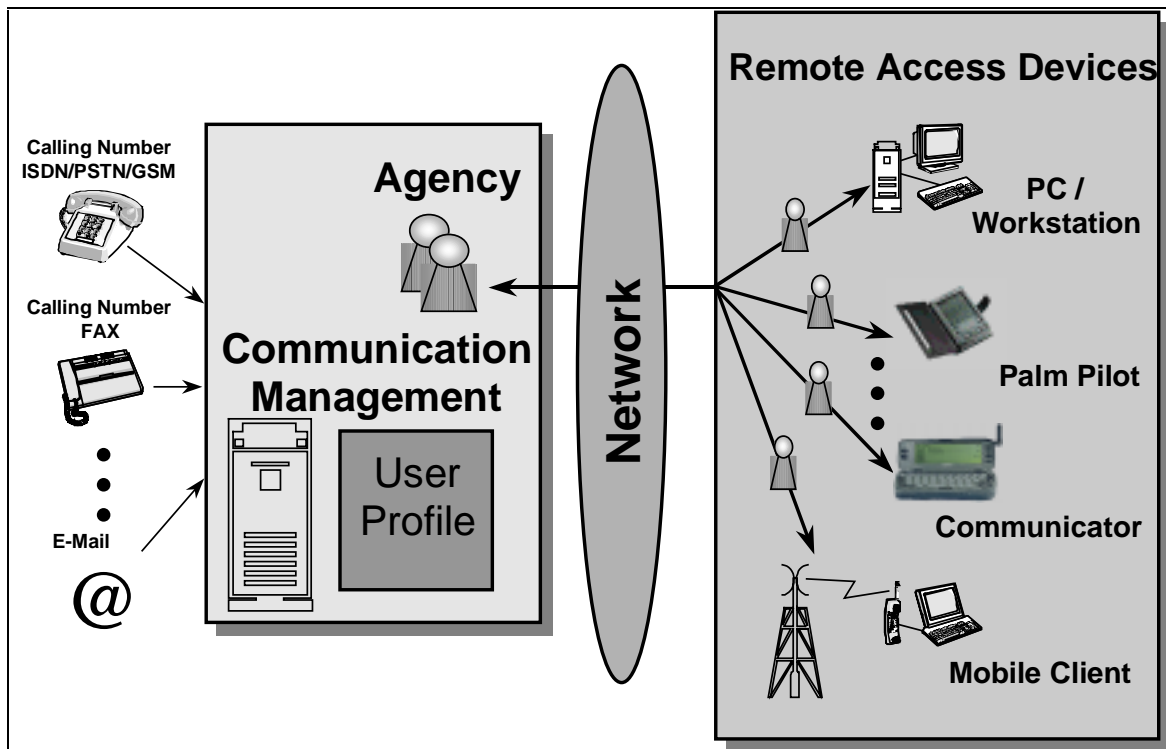


Figure 3: Adaptive Profile Management

- Remote, network and platform independent modification of all user profiles in a universal way,
- Handling of different end-systems, with different capabilities for profile management.

The APM enables the user to control his reachability for all his communication services depending mainly on the following conditions:

Time dependent conditions: The APM will enable a time dependent handling of all communication requests. For example, a UMTS user who attends a meeting or has an appointment at the same time each week could set a time condition to ensure that all his incoming calls are unconditionally forwarded to his voice box (e.g. between 10 and 12 every Tuesday).

Service dependent conditions: In the future a UMTS user will have one number for all his communication needs (e.g. phone, fax, email etc.). In such a scenario the user

may wish to dynamically specify how each type of incoming communication request should be handled, e.g. reject particular fax messages, or forward particular e-mails to another person according to the sender or/and subject of each incoming e-mail. The APM will provide a means of handling these service dependent rules. In addition the APM will be responsible for determining the capability of the end-system assigned to deal with the communication to ensure it is capable of supporting the service or to enable a “conversion” of the incoming communication to a format suitable for delivery to the specified device.

User (screening) dependent conditions: In an UMTS environment users will require more flexible communication screening functions. For example, a user may wish to specify a list of people for which he will be reachable and will expect all other calls to be rejected or forwarded.

The differences in the computing and communications capabilities of end-systems

will affect how the user's profile can be presented to the user. The challenge for the APM is to dynamically adapt the presentation (GUI) of the APM to the different end-systems the user has. In the context of the VHE concept three presentations will be realised. The first one will be a fully graphically oriented version to run on a workstation (terminal with graphic support). The second one will only present the content as text and/or with a limited graphical output and the third one will be a voice based user interface. The capability set of the user interface has to be adapted to the terminal capabilities.

In order to meet the requirements mentioned above, on the one hand, a local intelligent communication agent should organise and handle the incoming communication requests according to the conditions being held in user's profile (see Figure 3). On the other hand, a mobile agent will be sent to the end-systems to provide an adapted GUI enabling the user to modify his profile remotely from any point of any network, regardless of the used platform. The user will also be able to use this graphical user interface in order to be informed remotely about the last personal communication activities executed by his communication agent, e.g. to view the received faxes or emails and to play recorded voice messages [12].

The required Agents can be realised by JAVA using JTAPI (JAVA Telephony API), JNI (JAVA Native Interface) and RMI (Remote Method Invocation). This service focuses on the comparison of the agent platforms and is thus being implemented on both types of platform.

Other mobile agents will enhance this service, e.g. conversion agents and agents arranging a so-called "quiet conversation" by choosing template voice samples from a menu.

6. Virtual Address Book

Nowadays, telecommunication and IT users possess a number of communication devices, e.g. PCs, PDAs, residential phones, desktops and mobile phones, most of which contain or provide access to an address book service. The service offered is generally limited in usability or flexibility for even today's communication requirements. For example, mobiles generally have device resident memory where alpha-numeric names can be manually stored for standard E.164 telephone numbers, but do not have a means of accessing new details in real-time. A desktop PC may provide access to company employee names and telephone numbers via a platform specific application or more recently via an Intranet, but will require the user to launch a separate application for storing personal address book information. Communication between these applications on different devices is not currently possible. In today's scenarios the user has no choice but to replicate address book data across different applications and to deal manually with any data inconsistencies [2].

The Virtual Address Book (VAB) service will allow a UMTS user to use several different types of terminals to access and update user data. These terminals will have different degrees of computing and communication resources and may be located at geographically distant locations or may be used while roaming in the home or a visited UMTS network. It will be the responsibility of the agent architecture for this service to provide a consistent view of user data irrespective of the location, database, terminal type, and network configuration. The agent architecture will also have to deal with the problem of information consistency and will address the related aspects of conflict resolution, locking capability and shared access to address book information. One of the key challenges of this service will be to provide

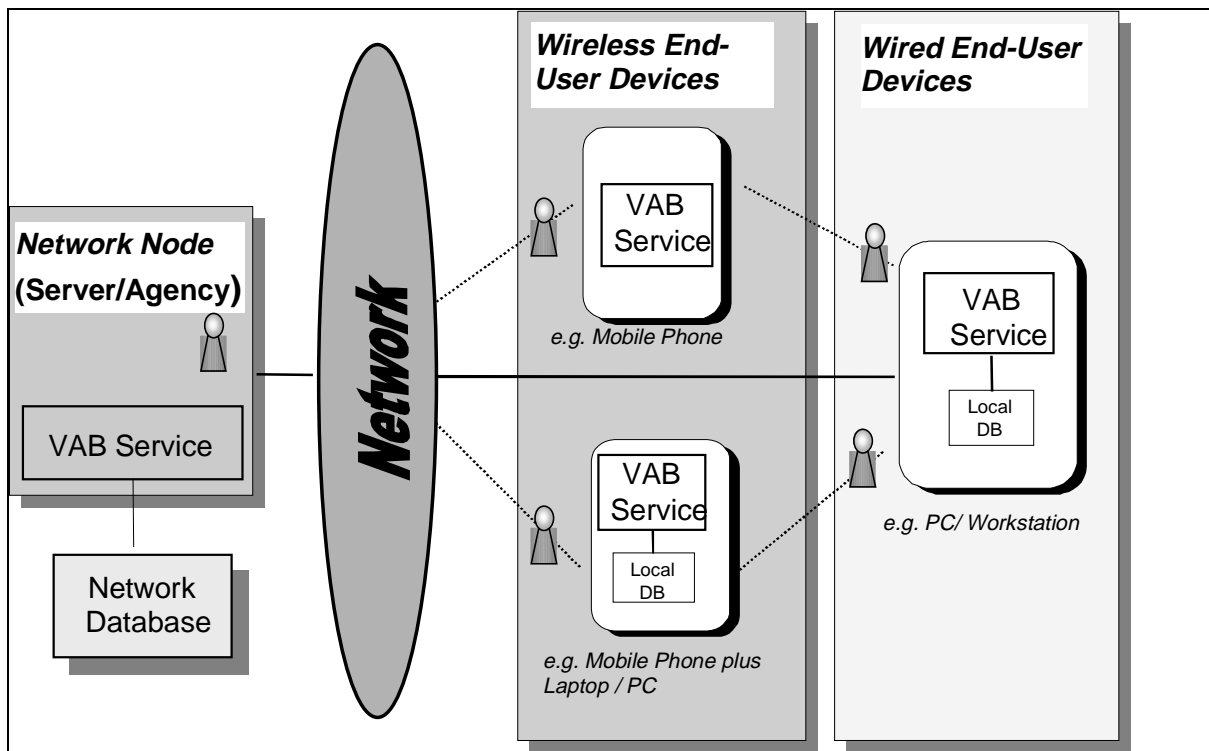


Figure 4: Virtual Address Book

an effective interworking between existing persistent-store/database. Mobile agents will be an elegant solution.

The architecture for the VAB service will consist of multiple collaborating agents to provide a flexible solution for the information consistency problem, which is the key issue for the VAB service. Introducing agents in this architecture provides additional value to the functionality offered by the persistent storage technologies that will be used to provide this service. This added value is achieved by the “intelligence” of the agents that consider user preferences, service features, terminal and communication characteristics. The key advantages in this regard are the following:

- A large distributed and heterogeneous system such as UMTS will use a variety of persistent storage technologies. Interworking between these technologies will be difficult. The database agents eliminate this problem by providing an optimised solution that is

achieved by collaboration between agents representing these technologies. For example, a user uses a file system to store information on his mobile terminal and an object-oriented (OO) database on his home PC. In order to get updated information on his mobile terminal the file system agents can provide adapter functionality that facilitates the negotiation with agents managing information in the OO database.

- An agent solution has a special advantage in wireless situations where bandwidth is scarce and costly. For example, a mobile user would like to get an updated version of his information but this would be costly while he is moving. With this architecture it is possible to initiate an update request and let the user agent collect information from different hosts off-line. In this scenario if the user needs an urgent access to some specific information the user agent can deliver this information as soon as the user

comes on-line. For non-urgent updates the user agent can deliver information when the cost to do so is within the user specified limits.

- In order to create a VHE at a foreign location, the user agent can collaborate with agents at this new location to personalise this new environment according to the user's preferences. This requires interworking between different persistent store mechanisms and information presentation environments to create a notion of the user's virtual home setting.

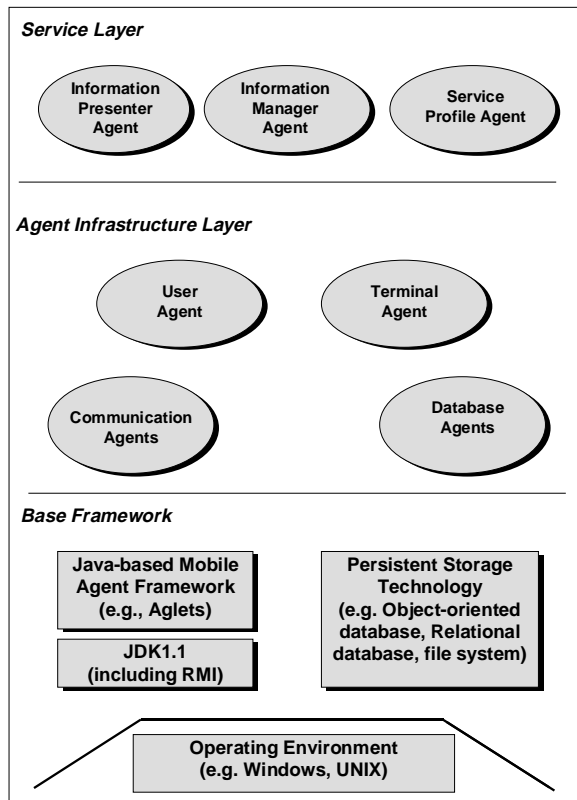


Figure 5: Information consistency agent architecture

Figure 5 shows a schematic view of the agent architecture. This architecture is based on a layered model in which agents representing the functionality of a service are placed in the *Service Layer*. Each service is represented by a *service profile agent* that helps in creating a VHE for that

service at any networked environment where this service is supported. An example task for this agent for the VAB service is to keep track of the host(s) with which this service must synchronise to get an updated user information. The *information presenter agent* considers user's preferences, terminal characteristics, and service profile to display a consistent view of the user information/data in an optimal fashion. It is the responsibility of the *information manager agent* to communicate with the *Agent Infrastructure Layer* to access and arrange to store user information/data to and from the persistent store (local or/and networked).

This layer consists of a service independent agent-based environment that provides generic functionality that any service can use to perform its task. For the VAB at least four classes of agents are required.

User agent: This agent manages user preferences (e.g. how frequently to synchronise information with other terminals/hosts) and plays an important role in creating the VHE for the user.

Terminal agent: This agent is responsible for managing the terminal characteristics (e.g. display type and size) and also plays an important role in creating a VHE. Terminal and user agents are consulted by other service and infrastructure agents to personalise/optimize tasks for the user.

Communication agents: These agents are responsible for managing communication with external agents, respectively applications. For the VAB an agent in this class could, for example, access the VAB to get a UMTS address for a user terminal. These agents that take care of communication tasks using different types of protocols and communication infrastructures.

Database agents: These agents are specialised to manage persistent storage issues (e.g. synchronisation, conflict resolution, information sharing) as identified

before. It is this class that provides information consistency for the VAB service. The information manager agent consults this class to perform information related operations (e.g. create a new record) for the VAB. The database agents provide a conceptual layer that mediates between a service and a persistent storage mechanism. This layer is reactive in nature, i.e. depending upon the data related task that needs to be performed, it initiates appropriate agents to facilitate and coordinate to accomplish this task. Database agents make the VAB service independent of the persistent storage technology used in a terminal.

The service and agent infrastructure levels are layered on top of a software environment (i.e. a base framework) for the terminal device. For this prototype, this environment is structured to provide a JAVA-based mobile agent framework (e.g. Aglets or Voyager, see section 2.2). The persistent storage mechanism used could vary from terminal type to terminal type.

This architecture reflects a peer-to-peer scenario. For a client-server configuration the agent infrastructure layer will be split and some components will be moved to the server (network) side. Depending upon the functionality that the server side needs to handle most of the database agents are relocated at the server side to handle requests from multiple clients. The server architecture needs also to be equipped with additional functionality (e.g. load balancing, high availability, fault tolerance) to handle multiple request from different client terminals. One of the biggest advantages of mobile agents in this architecture is that this relocation of infrastructure layer agents at the server side could be achieved at run time to provide adaptive system configuration, a key characteristic for achieving a VHE.

7. Conclusion

The service investigations lead to the conclusion that the VHE concept can be best implemented by using Agent Technology. Furthermore the combination of the VHE concept and Agent Technology allows easy and rapid development of new sophisticated teleservices. The feasibility of the approach is being tested within the European ACTS project CAMELEON, where the described services as well as a so-called Flexible Financial Service are being implemented and proved on two platforms allowing comparisons and the development of interoperability as well as migration strategies.

The services described as well as the concept of the VHE show a high potential of becoming commercially relevant.

8. References

- [1] ETSI TS 22.01. *Universal Mobile Telecommunication System (UMTS): Service aspects; Service principles*. UMTS Draft Version 3.1.0, July 1997.
- [2] ACTS Project Proposal #30001. *CAMELEON*. CAMELEON Consortium, September 1997.
- [3] Mouly M., Pautet M.-B. *The GSM System for Mobile Communications*. Published by the authors, 1992.
- [4] ETSI TS 22.70. *Universal Mobile Telecommunication System (UMTS): Virtual Home Environment*. UMTS Draft Version 0.0.3, June 1997.
- [5] Magedanz T., Popescu-Zeletin R.. *Towards "Intelligence on Demand" - On the Impacts of Intelligent Agents on IN*. 4th International Conference on Intelligence in Networks, Bordeaux, France, November 1996.
- [6] JAVA, <http://java.sun.com/>
- [7] IBM Aglets, <http://www.trl.ibm.co.jp/aglets/>

- [8] Voyager,
<http://www.objectspace.com/voyager/>
- [9] OMG, <http://www.omg.org/>
- [10] FIPA, <http://drogo.cselt.stet.it/fipa/>
- [11] Agent Society, <http://www.agent.org/>
- [12] Kluwe S. *Intelligent Communication Manager: Conception and realisation of a platform-independent remote access over the Internet*, Master Thesis, Communication Networks, RWTH Aachen, November 1997.