MOBILE AGENTS FOR MOBILE USERS

[1] Do van Thanh, [2] Sverre Steensen & [3] Jan A. Audestad
[1] Ericsson Norway Applied Research Center, P.O Box 34 N1361 Billingstad, Norway - etodvt@eto.ericsson.se - Tel: +47 66 84 12 00
[2] University of Oslo - UniK P.O Box 70 N-2007 Kjeller Norway - sverrest@ifi.uio.no - Tel: +47 63 81 45 70
[3] Telenor AS P.O Box 6701 St. Olavs plass N-0130 Oslo Norway – jan-arild.audestad@telenor.com – Tel: +47 22 77 99 52r

Key words: Mobile Agents, Mobile code, Mobility, Distributed Object Computing, Open Distributed Computing, Mobile Computing, Telecom and Computing convergence

Abstract: In this paper we propose the use of mobile agents in the implementation of mobility support. The paper starts with a presentation of the mobile agent concept. Then, the notion of ideal user mobility is presented and followed by the current mobility support approaches. The three mechanisms used to support mobility, namely channel establishment, pre-duplication and dynamic duplication are then explained. It is then shown that the mobile agent concept is probably the most suitable concept to realize mobility support since it allows a dynamic combination of all the tmenitoned mechanisms. The paper concludes with some comments regarding the use of mobile agent concept and suggestions for future studies.

Background: The work described in this paper results from the mobility part of the TELEcom REsearch Programme TELEREP, carried out at the Ericsson Norway Applied Research Center (NorARC) in collaboration with the University of Oslo, Center for Technology at Kjeller UniK [1]. The programme has been established to obtain practical experience with Distributed Object Computing methods such as ODP/TINA [2] [3], [4], CORBA [5], mobile codes in the implementation of TMN and IN functionality, and to study how mobility and security could be provided in the DPE environment.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35581-8_35

T. Yongchareon et al. (eds.), *Intelligence in Networks*

1. INTRODUCTION

The traditional approach to the specification and design of telecommunications services based on proprietary solutions and with low level of modularity does not meet the new needs. New technologies are required to facilitate the introduction of new services or the adaptation of the current ones. Fortunately, the convergence in technology enables the use of concepts, principles and methodologies from the computing world. It is therefore not surprising that concepts, principles and platforms defined in Open Distributed Processing (ODP) and Distributed Object Computing are adopted and used in the telecommunications systems, bearing in mind that these systems are probably the largest and most complex of all existing distributed systems. The main idea is to consider the distributed system consisting of heterogeneous nodes as a huge virtual mainframe. All the functions and mechanisms necessary to support distribution are hidden to the applications by the introduction of distribution transparencies. The defined transparencies are access, location, migration, relocation, replication, persistence, transaction and failure. An open distributed platform must at least offer access and location transparencies such as in the case of CORBA or Java.

However, there is a different concept called **mobile agent**, which does not provide location transparency to the applications. Indeed, the applications should exploit their location awareness to improve the performance. Mobile agent is actually not a new concept but is gaining more and more popularity with the growth of the Internet since the concept is intended for huge distributed systems such as the Internet, although it is not yet used on that scale. Although Open distributed systems provide process or object migration, it is designed with smaller scale computer networks in mind and thus assuming high bandwidth, small predictable latency and trust. The mobile agent concept addresses instead large scale setting where networks consist of heterogeneous hosts managed by different authorities with different level of trust, and connected by links with different bandwidths ranging from limited wireless connections to high speed optical links. This paper presents the approach of using the mobile agent concept in the design and implementation of of mobility support.

2. MOBILE AGENT CONCEPT

It is not trivial to provide an unambiguous and unanimous definition of an agent. We choose to adopt the basic agent model defined by [6]:

"A computational entity, which acts on behalf of others, is autonomous, pro-active and reactive, and exhibits the capability to learn, cooperate and move"

Autonomy implies that an agent should be able to operate independently from its user. Hence, an agent should be able to make rational decisions based upon the information it has gathered and the knowledge of its own task and purpose. **Pro-activeness** implies that an agent should take initiative in order to achieve its goal. **Reactivity** is the ability to sense and respond to changes in the environment.

We define a mobile agent as *a software agent that can move between locations in a heterogeneous network.* In addition to the basic model, a mobile agent is characterized by:

- a **life cycle model** defining the framework to create, destroy, start, suspend, stop dispatch and retract agents.
- a **computational model** referring to the computational capabilities of an agent, which include data manipulation and thread control primitives.
- a **security model** limits the ways an agent's ability to utilize the network and vice versa.
- a **communication model** focus on agent to agent communication and communication between an agent and e.g. the network. This can be based on simple message passing scheme or more complex approaches like KQML (Knowledge Query Management Language) or ACL (Agent Communication Language) proposed by FIPA (Foundation for Intelligent Physical Agents).
- and a **navigation model** gives the requirements to the issues of transporting an agent between two computational entities and of the naming and identification of agents in the system.

In order to build applications based on mobile agents that can be used on a large scale, the system must comprise a mobile agent framework, which supports all the mentioned models. An application is modeled as a collection of agents, each agent occupying different places at different times since it can move from one place to another. There is a diversion in framework concept. Some frameworks such as Aglet, abstracts everything in the system by the unique concept of agent. Some others such as Voyager allow a combination of agent and object concepts. Indeed, agents are viewed as extension of traditional objects. While the first approach yields a more coherent system that is easier to maintain and upgrade, the second approach enables easy adoption since migration from existing object system to agent system can be done gradually.

It is natural that the introduction of mobile agent framework will incur overheads in processing, memory demands, access response and execution time, etc. However, thanks to rapid and considerable progress of the underlying technology the mentioned overheads are no longer insurmountable obstacles. Indeed, the most commonly used technology Java can be quite small and fit for embedded systems. In addition, the future arrival of Java chips and Java system extension like Jini will allow full deployment of agent based applications with acceptable performance.

3. IDEAL SITUATION FOR A MOBILE USER

To illustrate the feasibility of the mobile agent concept we are going to investigate how a strategically mobile agent may benefit a mobile user. When a user moves to a foreign host and after authentication and authorization he would often like to access the system according to his own user profile, i.e. having the same documents and applications, the same icons, the same look and feel, the same environment as at his home site.

Ideally a user should be able to access the same applications/services and all his files or documents in exactly the same way, in the same presentation, with the same look-and-feel that he has personalized, no matter when and where he is or no matter what terminal he is using. Indeed, ideally the technology should be able to adapt to the user instead of requiring that the user must adapt to it, as the situation is today.

As depicted in *Figure 1*, ideal user mobility means the ubiquitous availability of the user's applications, his data files and his user profiles independently of the terminal in use. The **applications** comprise both the communication ones such as voice phone, video phone, conferencing, etc and the computing ones such as word processor, spreadsheet, mail, specific software, etc. The applications can use one medium or multiple media such as voice, video and data. The **data files** are of all types, either structured documents or unstructured raw data files. They can contain data for different types of media such as text, sound, picture or video. The **user profiles** capture the preferences, subscribed services and personalization that the user has chosen. Each user may have several profiles, for example one main profile for the main presentation and one for each application.

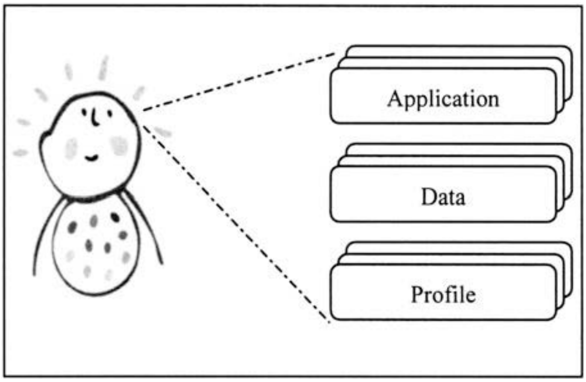

Figure 1. Ubiquitous availablity of applications, data files and user profiles

However, the physical characteristics of the underlying network and the terminal such as display size, processing and storage capability, media type, etc. are the unalterable limitations that prevent the total realization of the described ideal situation. Fortunately, the user usually understands such limitations and he is willing to accept some degradation or alteration. But, he has more difficulty to accept that given the required connectivity and a terminal with the required capability he is not able to access his applications, his documents. It is neither acceptable to require that the user must have quite a lot technical knowledge or must perform cumbersome, and time consuming configuration tasks.

In order to have a clearer view on how mobility is supported today, let us consider successively the cases of GSM (Global System for Mobile communication), UPT (Universal Personal Telecommunication), Mobile IP and Telnet/Remote login.

4. GSM

GSM (Global System for Mobile communication) offers terminal mobility meaning that the mobile user get the access to his services through his mobile cellular phone. GSM has telephony, i.e. voice communication as main application or basic service and a few other applications called supplementary services. The applications/services are well defined and complies with the ETSI (European Telecommunications Standards Institute) specifications. The application software (also called logic) must be installed and run on every location that supports telephony and supplementary applications. Each user has a user profile containing service restriction data such as list of subscribed services, roaming restriction, security data, charging data and routing data. The user profile is stored at the home site of the user, or more specifically in the Home Location Register (HLR). A copy is made and temporally stored at the domain currently visited by the user, or more specifically the Visitor Location Register (VLR). GSM offers very limited storage capability for user data. The existing user data are voice messages or SMS (Short Message Service) messages, which are saved in the home domain.

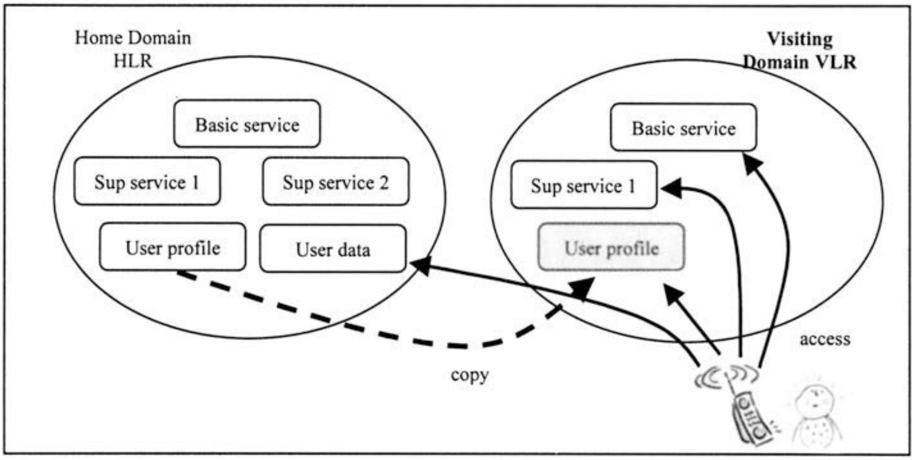

Figure 2. Mobility support in GSM

Figure 2 shows a user moving from his home domain (HLR) to a visiting domain (VLR) with his phone. In the visiting domain there is installed only logic for basic service, i.e. telephony and supplementary service 1 but not supplementary service 2. Supplementary service 2 is hence not available for him during his visit at the visiting domain. His profile is copied over to the visiting domain and made available for him. His data remains at his home domain and can be accessed through a channel between the visiting domain and his home domain. GSM is functioning very well today when the number and type of applications are relatively small. But in the future with the blooming of applications in number and type combined with the variety in user's preference and choice, it is not rational to expect that all applications will be present at every visiting domain. The user will as a result not be able to access all his applications. Another problem is related to the limited amount of user data that is in disposition of the user. It is reasonable to expect that the user will require more space for his data.

5. UPT

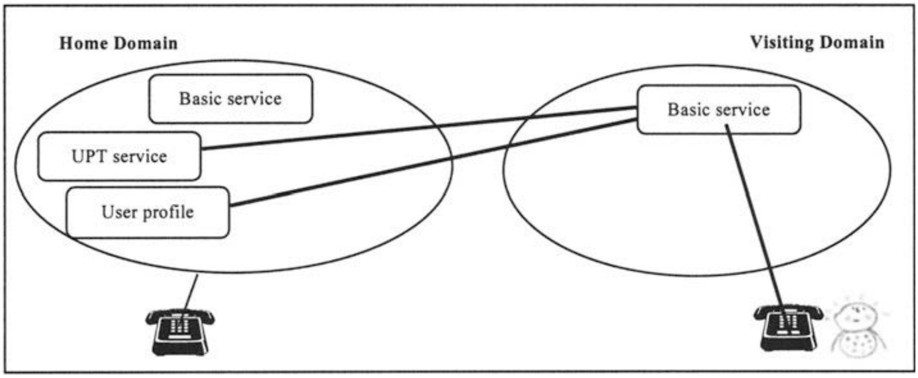

Figure 3. Mobility support in UPT

UPT (Universal Personal Telecommunication) allows the user to make and receive telephone calls from any terminal, fixed or mobile using a unique telephone number called UPT number. UPT can be viewed as a supplementary service built on top of the basic telephony service. Every time the UPT number is dialed, the UPT service is activated and performs the mapping of the UPT number to the number of the telephone that the user is currently used. Before making a call the user registers himself to the UPT service. The UPT service is only installed at the home domain of the user, more specifically in the home Service Control Point (SCP). The user profile is also kept only in the home domain, either in the SCP or the Service Data Point (SDP). UPT does not offer any data storage for the user.

Figure 3 shows a user using a telephone at a visiting domain. To receive phone calls, he registers to the UPT service at home by calling a green number from the visiting domain. Every call request addressed to him arrives to his home UPT service, which deduces the number of his current terminal. Connection can then be established to this terminal. To make a call, the user dials again the green number of his home UPT service and then enters the number to call. Connection is then prolonged from his home domain to the requested terminal

Although the UPT concept is valuable for the user it is today not implemented very efficiently since everything must first come the user home domain before going to his final location. A solution is to deploy the UPT service everywhere but this requires standardization, agreement and collaboration as in GSM.

6. MOBILE IP

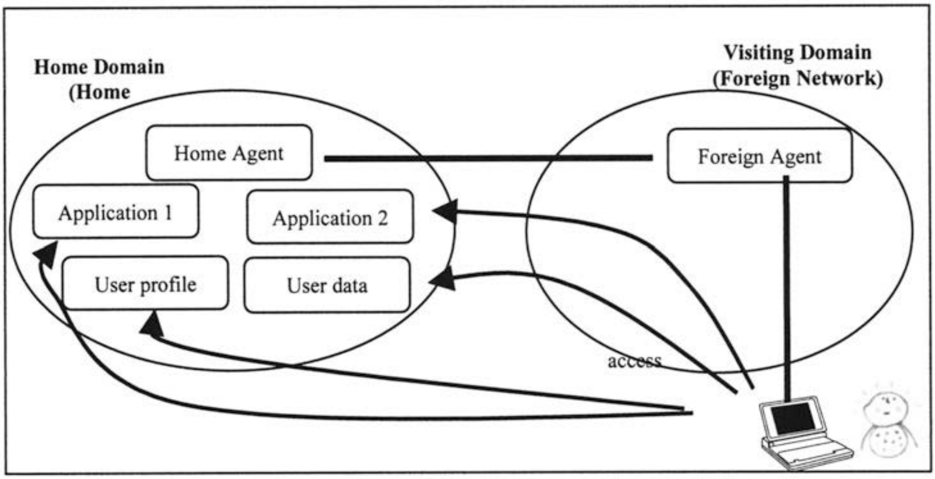

Figure 4. Mobility support in Mobile IP

IP or Internet Protocol is intended for networks of stationary computers or hosts. Each host is permanently assigned to an IP address that contains both the subnet identity and the host identity. There is no support for mobile hosts. The goal of mobile IP is to remedy this problem by allowing hosts to move and still be in communication with other hosts. By communication it is meant that other hosts are still able to send messages or packets to the mobile host and conversely the mobile host is able to send packets to other hosts. Mobile IP allows a mobile host to use two IP addresses. The *home address* is static and denotes the home domain, also called *home network* in mobile IP terminology. The *care-of-address* is dynamic and changes accordingly to the visiting domain, also called *foreign network,* to which the mobile host is currently attached. Whenever the mobile host is not at home, the *home agent* gets all the packets destined for the mobile hosts and forwards them to the foreign agent. The foreign agent arranges to deliver them to the mobile host.

Mobile IP focuses only on providing message communications for the mobile host and does not support the availability of other user applications, user profile or user data. Of course, if everything is stored locally on the mobile host then it will not be a problem. However, it is quite common for a business user to have only a subset of his applications, his data and profile on his laptop while the complete set remains on his corporate home domain. As shown in *Figure 4*, if the user wants to access applications, profiles or data that reside only at his home domain then he has to change some configuration parameters. This is a cumbersome, time-consuming and quite boring task which presumes that the user knows the parameters and theirs new values. The user must also remember the initial values of the parameters in order to be able to restore when coming home.

7. TELNET

In order to access his applications, his data and his profiles the user can also use Telnet, which has existed since the beginning of computer networks. As shown in *Figure 5* Telnet actually establishes a channel from the visiting domain to the home domain and allows the user to access his applications, data and profile located at his home domain. The Telnet protocol provides a standard interface for terminals and terminal-oriented communication. The protocol is normally used over TCP/IP but can also be supported by other protocols. When a Telnet connection is established each side is emulating a "Network Virtual Terminal" (NVT). It is this virtual device that provides the standard interface and hence dictates the look-and-feel. No personalization is possible. When establishing, options requests are sent back and forth as each side tries to negotiate the best possible services. The applications, which can be used from the visiting domain, are restricted by terminal resources, operating system and other restrictions at the application level. As a result, the user can normally access only a subset of his applications. Another disadvantage is the cost and trouble of keeping an open channel over a long period of time.

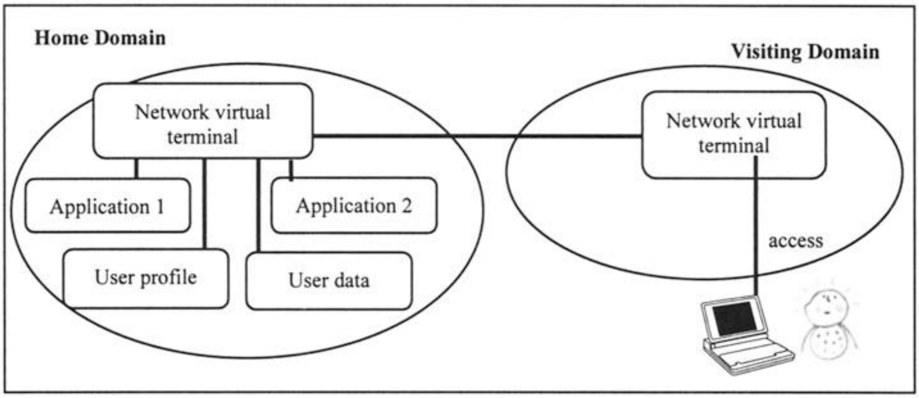

Figure 5. Mobility support with Telnet

8. IMPROVEMENTS BROUGHT BY MOBILE AGENT CONCEPT

From the cases investigated in the previous sections, it is possible to summarize the mechanisms used to implement the mobility support as follows:

1. Enabling access of user applications, profiles and data located at home: This is done by **establishing a channel** from the visiting domain to the home domain as in the case of UPT, Mobile IP and Telnet.
2. **Duplication** of user applications, profiles and data on the visiting domain: This mechanism has again two variants:
   a. Pre-duplication: This is done in the case of the basic service and some supplementary services in GSM
   b. Dynamic duplication: This is done in the case of the user profile in GSM. Moving code or logic is quite restricted until the emergence of Java.

All the three mechanisms1, 2a and 2b have both advantages and disadvantages, and are suitable only for a definite application, a definite situation, a definite visiting environment, a definite usage, etc. It is hence not possible to conclude that one mechanism is generally superior to another. Obviously, a combination of all three mechanisms would be the best solution. Further, such a combination should be dynamic, i.e. could be changed according to the situation. From this observation, the mobile agent concept is very suitable to implement mobility support since it allows all three mechanisms and also all possible combinations of them.

In fact, a strategically mobile agent might be the best solution to the mentioned problem. By strategic we mean that the agent will move according to the user preferences, network load and local resources. At a visiting domain, the user makes a request to his user agent, which is originally at his home domain. The user agent can inquire the visiting domain to obtain resource and capability information available on the visiting domain. The user agent can then travel to the visiting domain bringing along the user's profile. It will then initialize the foreign environment and the user's profile will be available to the user. Depending on different factors such as resources at the visiting domain, quality of the underlying network, type of applications, usage duration, etc. the user agent can decide which applications to move and which to access remotely. The combination used for each case will then be the most optimal according to the criteria defined by the user, the user agent and the visiting domain.

9. AN AGENT-BASED SYSTEM FOR MOBILITY SUPPORT

In this section we present the architecture of an agent-based system, which improves the support of mobility.

The whole system is composed of different sites. A site is governed by an administrator and can be composed of one of several networks (subnets) of nodes (hosts). Every site has a **SA** (Site Agent) assuming the administration function. It has the responsibility of giving access rights to the agents, garbage collect finished agents and make sure that the security model is not violated. It also has the responsibility to retract, i.e. to move a user agent according to the need of the user.

A user has subscription at a site. He is represented by a **UA** (User Agent), which acts on behalf of him and takes care of everything when the user is moving. The UA controls three tasks agents: the **AA** (Application Agent) in charge of the user applications, the **DA** in charge of the user data (document files) and the **PA** (Profile Agent) in charge of the user's profiles.

Figure 6 illustrates the case of a user X moving to a Visiting site. At his Home site, user X subscribes and has access to applications A_A and A_B. Application A_C is available at his home site but he does not use it. He has also the document files D_{X1}, D_{X2} and D_{X3}. His preferences are registered in the profiles P_{X1}, P_{X2} and P_{X3}. When visiting the Visiting site, user X needs all his applications A_A and A_B, the data files D_{X1} and D_{X3}, and the profiles P_{X1} and P_{X2}.

When user X logs in at the Visiting site. By his user name, which must be a unique identifier, the visiting site agent SA_V resolves his home site address and communicates with his UA_X. UA_X will perform the authentication of user X. If the authentication is successful, the DA_V retracts UA_X causing it to move to the visiting site.

When the UA_X arrives and is restored, it starts communicating with the SA_V, negotiating the access rights. The UA_X will find out what applications are already available at the host and how much memory and computing resources are available. It will then present this information to the user and suggests that the applications that the user normally uses that are not present, the users data files based on some criteria and the users profile are fetched. In our case, only application A_A needs to be fetched since application A_B already exists. Also, the data files D_{X1} and D_{X3}, and profiles P_{X1} and P_{X2} needs to be fetched. Based on the user's answer the UA_X retracts the task agents: AA_X, DA_X and PA_X from user X's home site.

When the task agents receive the retraction call, they will start to collect the files that are necessary to implement the requested services. These files might be spread at many different places at the user's home machine or on his local network. The agents then encapsulate these files and migrate to the remote host. In our case, AA_X takes care of A_A, DA_X of D_{X1} and D_{X3}, and PA_X of P_{X1} and P_{X2}. When the agents are restored at the remote host they distribute the files so the files will reside at the same directories as the user is accustomed to. If there are existing files with the same name, the agent will give that file a different name, and changes it back later when the user logs off.

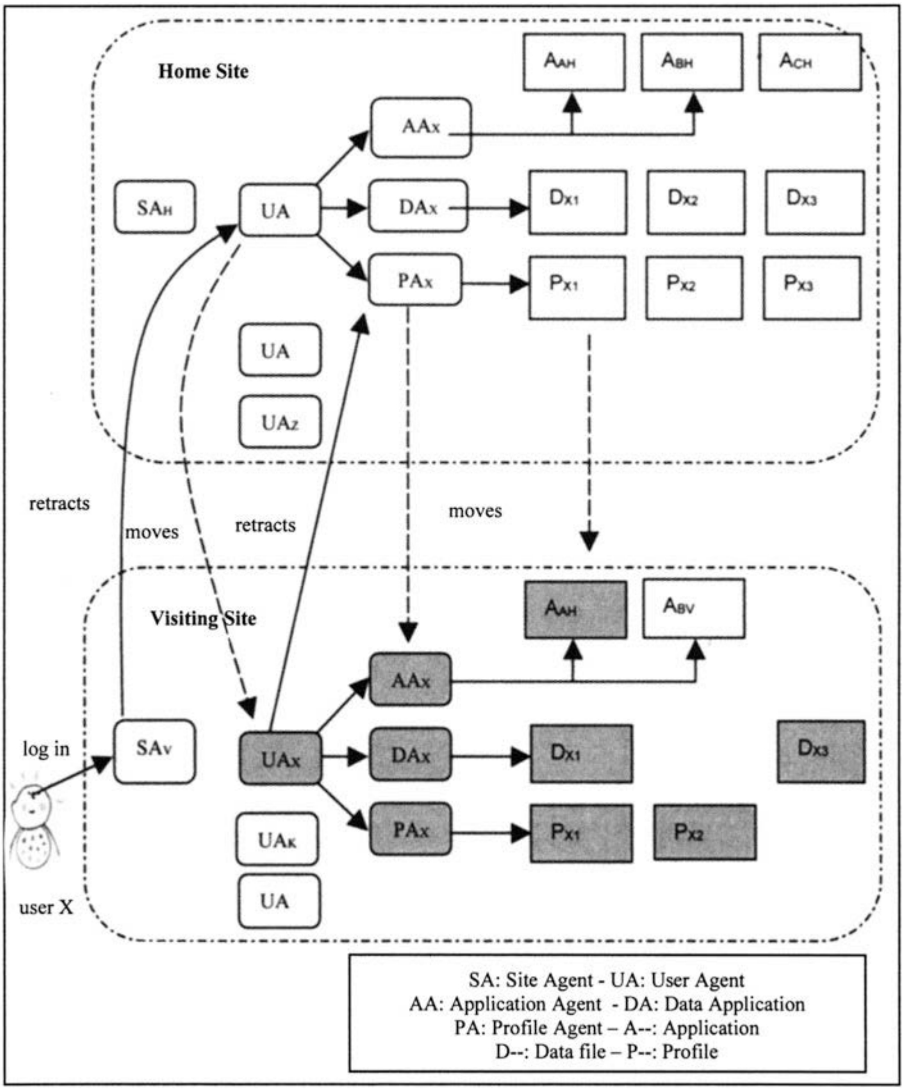

Figure 6. An agent-based system for mobility support

When user X logs off the SA_V will notify the UA_X. The UA_X will then ask the task agents if the files that were brought over at the migration have been changed, if that is the case those files must be returned to the home location before the files are destroyed at the remote site. The UA_X then kills the task agents, before it tells the SA_V that the UA_X has finished and lets the SA_V garbage collect it.

This approach will in our opinion lead to ideal user mobility. We could however achieve an even better result if the agents were strategically mobile. With that we mean that based on the speed of the network, the user's travel patterns and the processing powers at the visiting site the UA could make intelligent choices on how to best satisfy those needs. For instance when the UA migrates from the home site to the visiting site, it could measure the speed of the network connection. Based on this information the UA could indicate how long it would take to move the task agents. In another scenario the UA could ask how long the user is planning to work, based on information about the speed of the network, the UA could estimate the price of keeping an open connection (Telnet) or sending over the task agents. This would be a good solution if the applications were large and the user does not plan to work for a long time.

10. CONCLUSION

In this paper we have shown that the mobility support offered by current mobile systems are far from being ideal for the user and that many improvements can be achieved by using the mobile agent concept. Mobile agents seem to be a promising concept that has the flexibility and the adaptivness required in a mobile environment. However, as mentioned earlier, in order to be really usable, the mobile agent concept relies on the availability of technologies such as Java chip, Jini, etc. which can equalize the introduced overhead. In addition, there are several issues that also need to be resolved. One crucial issue is security. It is necessary to protect the visiting site from malicious agents and reciprocally, the agent from pirate sites. The user must also be protected so that his private files are not spread on different networks. Another issue also related to security is about firewall. The firewall of a site prevents any intrusion from outside and hence prevents also the visit of mobile agents. Another firewall scheme is needed. Last but not least, since the concept is still new, standards are still lacking despite of the efforts from OMG and FIPA. This can lead to incompatibility between different mobile agent systems.

REFERENCES

[1] Pål Spilling. The telecom research program telerep at unik. UNIK Aarsrapport 1996, 1996. http:www.unik.no/ paal/telerep.html.

[2] TINA-C. Overall Concepts and Principles of TINA, February 1995.

[3] ITU-TS. Basic Reference Model of Open Distributed Processing - Part 1 Overview and guide to use the Reference Model. Rec.X901(ISO/IEC 10746-1).

[4] ITU-TS. Basic Reference Model of Open Distributed Processing - Part 2 Descriptive model. Rec.X901(ISO/IEC 10746-2).

[5] OMG. The common Object Request Broker: Architecture and specification – Rev 2.2 February 99

[6] B. Pagurek A. Bieszczad and T. White: Mobile agents for network management, IEEE Communication Surveys, Sept 1998

[7] (http://www.objectspace.com/)

[8] (http://www.trl.ibm.co.jp/aglets/)

[9] (http://www.cs.dartmouth.edu/~agent)

[10] Crystaliz Inc, General Magic Inc, GMD FOKUS, IBM, TOG: OMG Joint Submission "Mobile Agent System Interoperability Facility", November 1997, available via ftp://ftp.omg.org/pub/docs/orbos/97-10-05.pdf

[11] Asgeir Tomasgard, Jan A Audestad, Shane Dye, Leen Stougie, Maarten H van der Vlerk and Stein Wallace: Modelling Aspects of Distributed Processing in Telecommunication Networks, Annals of Operations Research 82 (1998)161-184.