MOBILE AGENTS FOR PROVIDING MOBILE COMPUTERS WITH DATA SERVICES *

Yolanda Villate David Gil Alfredo Goñi Arantza Illarramendi

Departamento de Sistemas y Lenguajes Informáticos.
Universidad del País Vasco.
Apdo. 649. (20080) Donostia (Spain)

Abstract

Nowadays, the interest in mobile computing is growing and many issues related to it are aim of research. In this paper, we propose a system that provides mobile computers with new data services, such as renting data lockers, Internet a la carte, etc. The basic idea behind the system is to combine the use of the indirect model and the mobile agent technology. According to the indirect model our system incorporates several functions and services in an intermediary element that relieves the mobile computers from many tasks and increases their capabilities. Mobile agent technology may reduce the workload in the mobile computers by defining agents that represent them and that can be executed in the intermediary element and in other places. We present in this paper an overview of the functional and operational features of the system.

Keywords: *Indirect Model, Mobile Computing, Mobile Agent Technology*

1 Introduction

The combination of wireless communication networks and portable computers allows a new kind of computation: mobile computing. Although the wireless communication networks were designed for the transport of voice signals, their use for data transport is growing and is expected that in the 2000 year, half of the information flow in GSM networks (Global System for Mobile communications) will correspond to data transmission.

In [IB94] it is presented the widely accepted architecture of a system that supports mobile computing. This model consists of the following kind of components: *Mobile Units* (MU), *Base Stations* (BS) (also called *Mobile Support Stations* (MSS), that are fixed hosts augmented with a wireless interface), and *Fixed Hosts* (FH) without wireless interface. Moreover, it has also been suggested the indirect model in [BBIM93] in order to increase the throughput in mobile computing. The basic idea of that model is: whenever the interaction between two computers takes place over two radically different media, like

*This work is supported by *CICYT: Comisión Interministerial de Ciencia y Tecnología*, Spain. [TIC97-0962]

wire and radio waves, their interaction is broken down into two phases, one for every kind of media. An intermediary element, called *proxy*, is placed in one point between the two computers. That proxy manages the interaction among the computers taking into account the different nature of the two media involved. The proxy tries to relieve the more limited extreme of the communication from some tasks, but its existence can even remain unnoticed for the two computers. Taking into account the philosophy behind the indirect model, we propose to extend the previously mentioned architecture with a new element called *Gateway Support Node*[1](GSN).

The GSN would be situated in the border between circuit switching and packet switching, more than in the border between wired and wireless. The number of GSN hosts in the network depends on the geographical coverage and the number of users of data services. On this model every BS has a GSN as 'tutor' or manager, and one GSN manages one or more BSs, whose cells usually constitute a geographical area (Figure 1). In that GSN we incorporate services and facilities with the goal of relieving the mobile computers from many tasks and increasing their capabilities, respecting their natural limitations. Those services and facilities should be offered by the communication company owner of the cellular network.

The use of the agent technology when implementing those services and facilities provides the following advantages: a) *asynchronous* communications where the elements involved into the communication do not have to be connected all the time. In this case the MU may decide to be disconnected while the agent is communicating with the other element. This may be interesting for example when accessing databases where transactions can take a long time. b) *autonomous* communications where the agent has more knowledge about the element of the communication to which it represents and may take some decisions on his/her behalf. This may be interesting for example when accessing databases and some transactions fail. c) *remote* communications where the agent can make use of remote facilities or resources: memory, CPU, less restrictions with bandwith, etc. We could have also implemented our system by using more traditional techniques such as Client/Server and standard message technology. However, the previous advantages have convinced us to use agent technology. In particular, we are using mobile agent technology and although there is a time needed until mobile agents cache on, we believe that the mobile agent technology is going to grow in popularity and application in the near future. In our proposal, mobile agents are used to represent the MU on the GSN, and for carrying out from the various tasks on behalf of the MU. Mobile agent technology, apart from the previous advantages, allows us to migrate processes among different machines.

In the rest of this paper we present first the functional architecture of our proposal and how some problems related to the mobile computing framework are considered. Next we show an outline of the operational behaviour using the agent technology.

[1]The Gateway Support Node name is borrowed from the General Packet Radio Service (GPRS), a similar service to the Cellular Digital Packet Data (CDPD) that is under definition for second phase of the GSM standard. We take GSM and GPRS as cellular network model for our work.

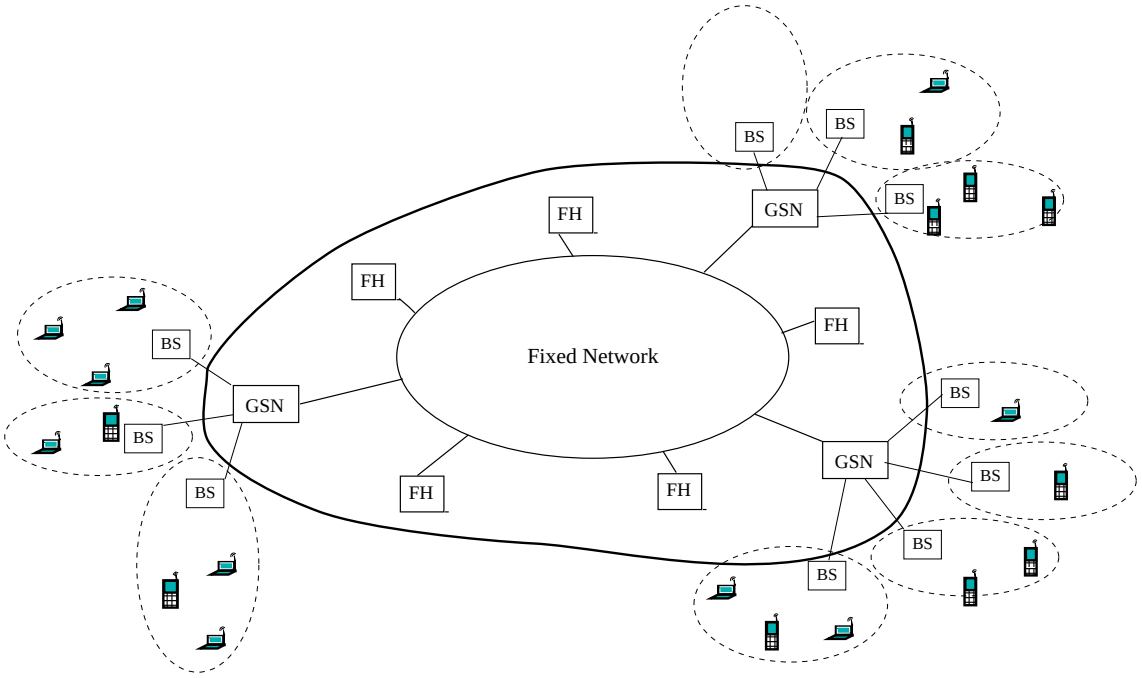

Figure 1: Mobile Computing environment.

2 Functional Architecture

In this section we describe the features of the new element introduced into the architecture, the GSN and the features of the mobile units.

2.1 The Mobile Unit (MU)

In the MU, new features with the goal of controlling and managing available resources in a transparent way for the user are included. There is also an interface that facilitates the access to all the services offered through the GSN (explained in the next subsection) to the mobile user, and a local cache, as proposed in other works [BI95, SL97].

2.2 The Gateway Support Node (GSN)

The GSN is the intermediary element in the communications between the mobile units under its coverage and all other hosts of the network (mobile or fixed). The pair GSN+MU will dissemble that the MU behaves like a fixed computer to the rest of the network, i.e., the GSN lends its identity to the set of mobile units that monitors, so when the GSN receives messages and data sent to the mobile units distributes them to the suitable MU. The GSN has a control about mobile units under its coverage (their identification, profiles, groups in which they can be included, etc.)

The GSN supports the following main functions and services:

- **Broadcast transmitter.** Broadcast can be used for disseminating general interest

information such as local traffic condition, weather forecast, financial information, etc., and those data asked for most of the mobile units.

- **Yellow pages.** It provides access to different data repositories with local general interest information, such as local restaurants, hospitals, hotels, etc.
- **Access to Internet a la carte.** The use of *push* technology to access to information channels would be too expensive for the mobile units; the GSN can do it for them. With this functionality the GSN provides mobile users with information about subject of their interest.
- **Available software for mobile units to use**, such as freeware software, software for rent, or filter libraries. This software could be used on the MU or on the GSN on behalf of the MU.
- **Temporal caching of data** for the MUs, and for the common use of the applications.
- **Rent of lockers in the GSN.** By renting some space in the GSN to which we call lockers, the user would have services such as: confidential access protected with a secret key to a locker, redirecting his/her email to a locker, maintenance of consistency of data stored in the locker, and so on. The rent of lockers can be made from private users or from corporate users, this could be the case of a company that rents a locker for its mobile workers (equipped with mobile units), that could use the locker for sharing information in a safe and private way.

2.3 Problems in the Mobile Computing Framework

In the following we show how recognized problems in the framework of mobile computers are considered in our proposal, namely: scarce bandwidth, limitations on the size and capabilities of mobile computers and battery power limitation.

- *The scarce bandwidth of wireless networks.* There are two main communication paradigms: point-to-point, direct communication between GSN and MU, and broadcast paradigm, to disseminate data to multiple mobile units. Algorithms for transmission and selection of information for broadcasting are studied in [IVB97, LS98], where broadcast is shown as a special cache on air, where there could be applied techniques and strategies similar to those ones studied for traditional caches. The use of broadcast saves bandwidth reducing the need of point-to-point communications between the MU and the GSN, and it also saves battery power of MU because it reduces uplink communications (MU to GSN), more expensive in terms of energy than downlink communications (GSN to MU). In our case the broadcasting will be planned in the GSN for all the area under its coverage, and then it will send specific orders and data to every BS for broadcasting.
- *Limitations on the size and capabilities of mobile computers.* These can constrain the human-computer interface; for example, it can be necessary to adapt the images

for their visualization reducing their size, definition or colors. For arrange data and communications to the specified characteristic of the MU we propose to use filters[2]. The majority of those filters are applied on the GSN with the goal of relieving the MU of some tasks. The use of filters on the GSN contributes also to save battery's power and wireless bandwidth, e.g. reducing the size or amount of data to transmit.

- *Battery power limitation.* Taking into account that the MU has a *majordomo* agent that represents it in the GSN, the MU is able to disconnect or be in doze mode for saving energy, while the agent works on its behalf, receiving data; processing them, maintaining data coherency with their sources, and so on. In the case when the MU goes to doze mode, this agent can wait during a specified time or until a specified event for waking up the MU.

3 Outline of the Operational Behaviour

In this section we present, for each component MU and GSN, the main modules that constitute them, and their behaviour from the point of view of the agent technology used for the implementation, but first we explain some general notions related with the agents.

Agent technology is not new on computer science [Mae97], it has been used for example on Artificial Intelligence, but mobile agents are a novelty. In general, an agent is a computer program that acts autonomously on behalf of a person or organization. Each agent has its own thread of execution so that it can perform tasks on its own initiative. A mobile agent is not bound to the system where it begins execution; it has the unique ability to transport itself from one system in a network to another. The ability to travel lets a mobile agent move to a system that contains an object with which the agent wants to interact and take advantage of being in the same host or network as the object. When an agent travels, it transports its state and code with it. An agent system is a platform that can create, interpret, execute, transfer and terminate agents. When an agent transfers itself, it travels between execution environments called *places*. A place is a context, within an agent system, where an agent can execute. This context can provide functions such as access control. The source place and the destination place can reside on the same agent system, or on different agent systems that support the same agent profile [Gro97].

3.1 Mobile Unit

The MU component is composed by three main modules (see figure 2). The MU Resource Manager that administrates all the resources available at the MU such as storage space, cache, power, network communications, and so on. The Place and Agent Manager that creates the place and the agent which works on the MU and manages them. And finally, the Mobility Administrator for Mobile Unit module that manages the mobility transparently to the user and the user applications.

From the operational point of view, first a place is created on the MU. Then, an agent called MU static agent is also created. This agent administers MU resources, decides which

[2]Filters are programs used for processing every message coming or going to the MU. They can abolish, delay, reorganize into segments or transform the message [ZD97].

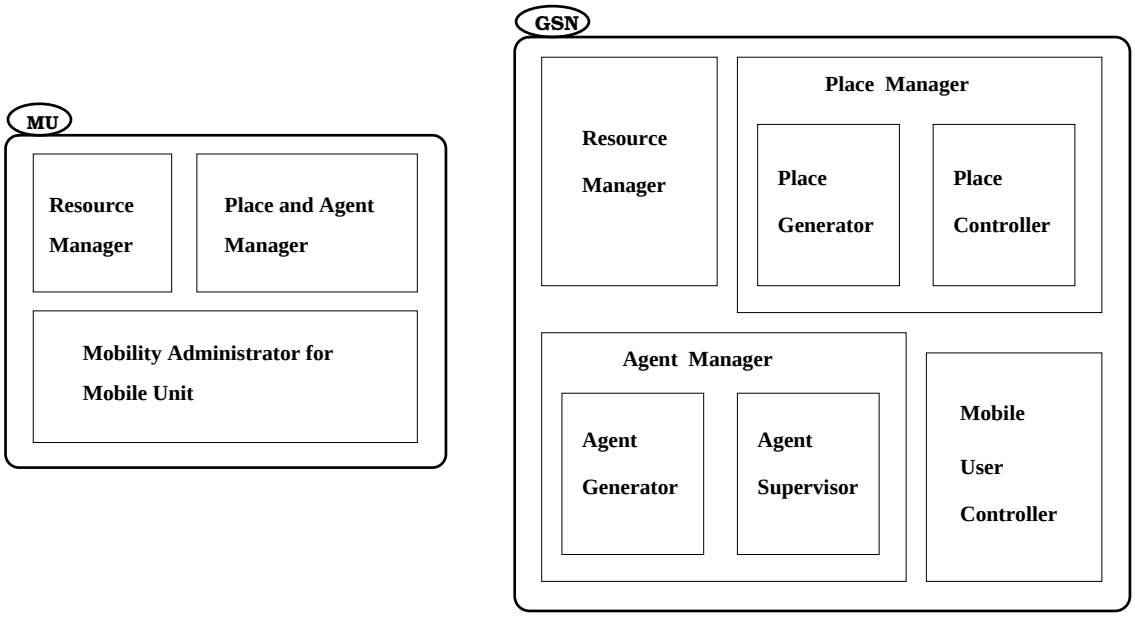

Figure 2: MU and GSN main modules.

filters must be applied to the information, sends queries and requests to the network when they cannot be answered with its own cache, maintains the consistency of that cache, guards the opened communications, etc. It also manages a profile of the MU with data about their characteristics, its identification and its user identification. This agent is active whenever the MU is working. The first time the MU needs to communicate with an outside element, the static agent creates a mobile agent called majordomo and sends it to the GSN - to the entry place called *Inventory Place* - where it represents to the MU. This couple of agents follow a design pattern called Master/Slave [AL98]. The communication between both agents takes place by sending messages from one to another. The static agent contacts the majordomo to give him new tasks, such as finding the answer to some query. The majordomo agent contacts the static one in order to inform it about the arrival of data or new email, and also in order to waken the MU (if it is in doze mode) and so on. The majordomo agent remains in the GSN until the static agent orders him to come back to the MU. Moreover, the majordomo agent changes from one GSN to another whenever the MU moves and changes of GSN.

3.2 GSN

In the GSN there are four main modules (see figure 2). The Place Manager, with its two submodules called Place Generator and Place Controller, is prepared to create and manage the different places, the interaction among places, and among the places and the rest of the system. The Agent Manager, with the Agent Generator and the Agent Supervisor, has a similar objective than the previous one, but concerned to the agents of the system. The Resource Manager which administrates the resources available at the GSN such as a second storage space, network bandwidth, cache, etc, and distributes them

among the different modules and places. Last, the Mobile Users Controller that manages all the aspects concerned to the mobile users, their connections and disconnections, current location and state, use of services, and so on.

The GSN is the front door for the MU to the network, the common point for all traffic created by or addressed to the MU. In addition of providing the functionalities that we have described in the previous section such as Broadcast transmitter, Yellow pages, etc., it must:

- administrate optimally the resources of base stations under its coberture (bandwidth, broadcast channels, number and list of mobile users).
- monitor users under its coberture (registering their localization, state, data, connection and disconnection time, profile, etc.)

Operationally, when the MU connects with a GSN, the MU sends the data necessary for its identification. Using those data the GSN must validate the MU identification and decide whether giving coverage or not. Once the MU gets coverage, it is ensured that the majordomo of the MU is placed in that GSN by one of the following ways: doing nothing if the majordomo is already there, bringing the majordomo from another GSN where it is or asking the MU to create a new majordomo and to send it to the GSN.

The GSN is able to interact with other GSNs with the goal of interchanging data, balancing the load of the system, etc. When a MU moves and changes from one GSN to another, the first GSN notifies the majordomo of the MU about stopping all processes and agents of the MU that were in execution on the GSN. The majordomo has to summarize the processes, encapsulate them with their data and carry all of them with it to the new GSN. Once in the new GSN, the majordomo restores itself and recovers the data, agents and processes for continuing their execution.

For the implementation, we have decided to define a place for every service or functionality that the GSN offers to its users and another one to which we call *Inventory Place* that constitutes the entry point to the GSN for the majordomo. Although it is true that we do not need to distinguish so many places and that it can be designed using only one, we think that our design has the following advantages:

- The system has more modularity. That gives a clear vision of its performance and more facility to manage it.
- The system is more robust and fault-resistant. Even when some places fault the system can still work.
- The location of the places is transparent, they can be in other GSN or in dedicated computers, so the system have more flexibility, one GSN does not need to support locally all the places or services.
- The system is more scalable, there can be added, actualized or eliminated places or services without drastic changes to the system.
- There is a great adaptability between places and the services that support. The system works better and reaches a higher throughput by creating places and agents specialized in giving a group of or a concrete service.

However, implementing it as a set of places, instead of as a unique place, increases the traffic of agents in the GSN, and apparently the use of resources is higher, but we think that the advantages mentioned before justify our decision. In the following we show the main features of the places.

- The Inventory place. It is not only the entry point to the GSN but also its nucleus. This place is called inventory because it contains information about the places and services available on the GSN where resides and in others. It also contains information about the users and BSs under the coverage of the GSN, needed for their control and administration. The Inventory is the place that receives and supports majordomos. Actually all the GSNs do not need to support services like Yellow pages, locker rent, Internet *a la carte* and libraries of software. With the inventory place a majordomo can do a transparent use of services, without worrying about whether the service is offered locally or in a near GSN.

- The Yellow Pages place. This place supports the Yellow Pages service that allows the access to several databases or data repositories located over the network. Access to these repositories is made by using the functionalities incorporated into the Yellow Pages place. These functionalities are designed in such a way that the agents do not have to know neither details of implementation of the repositories, nor the location of the repositories nor the communication mechanisms used. The only thing they have to know is the interface to make queries in that place[3]. When the MU wants, for example, to access to certain information of the *Yellow Pages*, the MU static agent sends a message to its *majordomo* specifying the query to the *Yellow Pages*. When receiving a message of that type, the majordomo creates an agent specialized in the *Yellow Pages*, to which it gives the query and orders it to go to the Yellow Pages place. Once in this place the agent accesses to the appropriate information source to obtain the requested data by using functionalities offered by the place, it receives the answer, and next it returns to the Inventory Place to give the answer to the majordomo. Once the majordomo receives the data, it takes charge of sending them, as soon as possible, to the static agent of the MU using a message.

- The Broadcast place. This place has a specialized agent reponsible for the broadcast policy for every BS under the coverage of its GSN, applying algorithms and techniques for deciding which data, when and how to send them. The agent of this place may need some information from the majordomos, Yellow Pages place, Internet *a la carte* place, etc., for deciding which data are demanded for most of the users.

- The Internet *a la carte* place. It is a place with agents specialized in the support, creation and access to information channels using push technology. When a user is interested in periodically receiving information about some subject, his majordomo creates a new agent, that is a specialist in Internet *a la carte*, and sends it to the place. The agent subscribes to some existing channels, or asks for the creation of a

[3]For the implementation we are using Aglets Workbench [Pag], using this system we can implement these functionalities as a set of objects, called properties, that we asign to every place.

new one, so that it can obtain the information and sends it to the majordomo with the desired periodicity.

- Available Software place. It is a place with agents specialized in searching, accessing and providing software available to users. Free-ware software, libraries of filters, software for renting with an associated rate will be offered in this place. This software could be downloaded into the MU, and run it in the MU, or it could be used and run in the GSN. When a majordomo needs to use a software, it creates a specialized agent. This agent goes to this place, searches, obtains the software and gives it to the majordomo, or in other cases the agent can run the software to achieve some task.

- The Cache place. This place maintains a cache for the common use of majordomos and applications. For example, when the MU requires some data (from remote databases, Web pages, etc.), its majordomo creates an agent which gets those data and stores them in this place. When the tasks finish sucessfully, the specialized agent informs the majordomo which returns the data to the MU. The period of time that those data can remain in the cache is limited. The MU can be disconnected while the data are being located.

- The Locker place. Some users need a dedicated space in the GSN, on this case they can rent a locker. Notice that in the cache place the data can be stored only for a short period of time. With respect to the Locker place, we have decided to model the compartments or lockers using a new kind of agents. These agents are specialists in the maintenance of lockers (in fact they are the lockers themselves). Such agents are created by the Locker place and assigned, each one of them, to a specific user or group of users. This solution allows us to manage the locker in a dynamic and flexible manner, creating the lockers at the same time that they are rented by new users; instead of choosing a model of physically separated compartments to implement the lockers.

  When the user wants to rent a locker, the static agent of the MU sends a message to its majordomo ordering it to rent a locker. When the majordomo receives this message it creates an agent specialized in the locker and then the majordomo sends it to the Locker place. When the Locker place notices the arrival of a such kind agent sended by a majordomo, the place creates a new locker agent and it presents them mutually, i.e., the place gives both agents a reference to the other, so they can interact together interchanging messages.

  The locker agent takes care of storing user data, maintains data consistency, saves the mail, processes results, communicates with the majordomo and assures the privacy and security of its locker. Every locker can be rented for a private user or for group of users, e.g., a company can rent a locker for its mobile workers equipped with mobile units, so that they can share information in a safe and private form.

  When the user agent created for the locker returns to the Inventory place, the asociated locker agent is destroyed and the space that it was using for data is liberated.

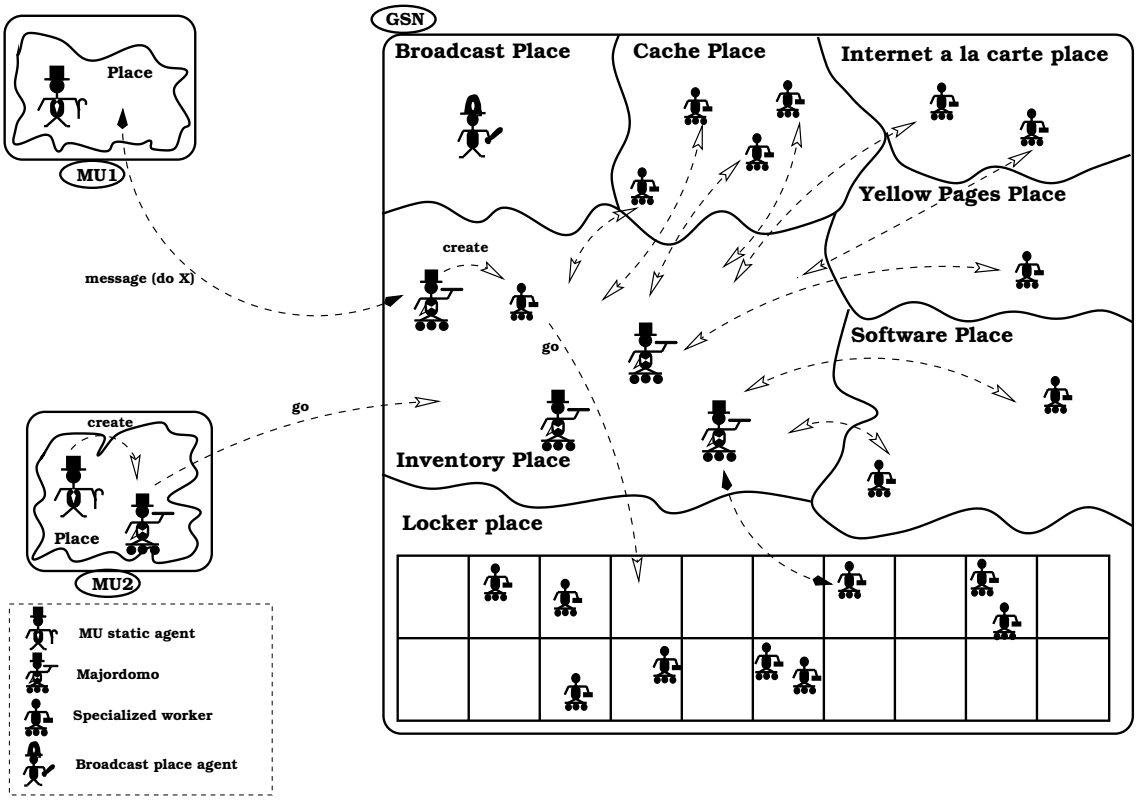

Figure 3: Places and agents in the GSN and in the MU.

In figure 3 we show the places, within some agents working in them, of the GSN and of the MU.

3.3 The Central System and the Lost Property Office

In order to give more functionalities to the system and specially for getting a secure system, two more elements are incorporated in the operational architecture of the system: the Central System and the Lost Property Office. These elements are distributed among several hosts (including GSNs and other hosts in the fixed network), owned by the communication company of the cellular network.

The Central System (CS)

The Central System is the liable element to manage and maintain all the system. It is its nucleus and the key stone for the security of the system.

We have said that the GSN must validate the MU identification. The information and mechanisms needed for user validation and authentication are at CS. There are also stored data about of location of the MUs, its state (off, on, doze), its last location, and so on. The CS maintains also the global catalogue of services for users and methods for accessing and using them, lists or groups of users and methods for registration. The CS

does statistics about use, access and so on of every MU and in general of the whole system.

The Lost Property Office (LPO)

In the mobile computing environment disconnections are very frequent, they can be caused for saving energy, or by going out of coverage or damage. When disconnections are for long periods of time (maybe days), the data of MUs and processes have to be summarized, encapsulated and sent to a new site (to which we call Lost Property Office) where they are stored until the MU reconnects and asks for them. The GSN reports it to CS who updates the state of the MU and its data. The main reason to have an LPO is to save space and resources in the GSN. If after some time, the MU reconnects, the CS reports to the involved GSN where MU data and processes can be found, so that GSN can recover and resume them. Data of MUs is saved in LPO for a long period of time before being definitively deleted.

4 Conclusions

In this paper we have briefly presented the main features of the system that we are building for increasing the mobile computer capabilities. The system is supported on two pillars: indirect model and mobile agent technology. The use of the indirect model allows us to relieve the mobile computers from many cost processing tasks and increase their capabilities (some laptop and palmtop have limited capabilities) by using the computing power provided by the intermediary element.

The use of mobile agent technology may allow us to save wireless bandwidth by reducing the communications needs and the amount of data exchanged. Moreover, the time that the mobile computer must be connected using the wireless media may decrease. The agent works in an autonomous and independent way and can react to events, allowing in this way to simplify the construction of robust and fault-tolerant distributed systems. Furthermore agents technology favors portability because they only depend on their execution environment and not on the computer platform and transport layer.

At the moment we are in phase of implementing a prototype for whose development we have selected the platform "Aglets Workbench" [Pag] that allows us to develop mobile agents in Java. The prototype is being implemented by modules to facilitate the incremental construction of the system. In the current state, four modules are being implemented: the MU place, the Inventory place, the Locker place and the Yellow Pages place.

After our first experiments, we have observed that dealing with GSM technology due to the scarce bandwith (maximum of 9600 bps), agent technology provides with a better performance when agents avoid wireless communications or when it is needed that the MU has to be connected for a long time in order to wait for the answer.

References

[AL98] Y. Aridor and D. B. Lange. Agent design patterns: Elements of agent application design. In *Proceedings of the Second International Conference on Autonomous Agents (Agents'98)*, May 1998.

[BBIM93] B.R. Badrinath, A. Bakre, T. Imielinski, and R. Marantz. Handling mobile clients: A case for indirect interaction. In *4th Workshop on Workstations Operating Systems*, 1993.

[BI95] D. Barbará and T. Imielinski. Sleepers and workaholics: Caching strategies in mobile environments (extended version). *The VLDB Journal*, 4(4), 1995.

[Gro97] Object Management Group. Mobile Agent System Interoperability Facilities Specification, November 1997. http://www.camb.opengroup.org/RI/MAF/.

[IB94] T. Imielinski and B.R. Badrinath. Mobile wireless computing: Challenges in data management. *Communications of the ACM*, pages 19–27, October 1994.

[IVB97] T. Imielinski, S. Viswanathan, and B.R. Badrinath. Data on air: Organization and access. *IEEE Transactions on Knowledge and Data Engineering*, 9(3), 1997.

[LS98] H.V. Leong and A. Si. Database caching over the air-storage. To appear in Computer Journal, 1998.

[Mae97] Pattie Maes. On software agents: Humanizing the global computer. *IEEE Internet Computing*, July-August 1997.

[Pag] IBM Aglets Workbench Home Page. http://www.trl.ibm.co.jp/aglets/.

[SL97] A. Si and H.V. Leong. Adaptive caching and refreshing in mobile databases. Personal Technologies, September 1997.

[ZD97] B. Zenel and D. Duchamp. A general purpose proxy filtering mechanism applied to the mobile environment. In *Proceedings of the Third Annual ACM/IEEE International Conference on Mobile Computing and Networking*, September 1997.