Data Services for Wireless Devices: from laptops to PDAs and from GSM to GPRS*

J. Rodríguez[1], Y. Villate[2], A. Goñi[1], A. Illarramendi[1], and E. Mena[2]

[1] LSI department, University of the Basque Country. Spain
{jibroarf,alfredo,jipileca}@si.ehu.es
http://siul02.si.ehu.es/
[2] IIS department. University of Zaragoza. Spain
{yvillate,emena}@posta.unizar.es
http://siul02.si.ehu.es/

Abstract. In this paper, we present briefly three new data services for wireless device users and the environment where they are defined. The implementation of those services is based on the use of the Client / Intercept / Server model and the agent technology. Moreover we show experimental results associated to the use of one of them that measure the influence of three different variables: the mobile computing model, the wireless network, and the kind of wireless device.

1 Introduction

Nowadays, the widespread use of wireless devices that can be connected to Internet allows users to access information from anywhere and at anytime. However, the management of Internet data services makes users of wireless devices face certain problems such as the unstability of the used media, the lack of suitability of existing services to the limited capabilities of some wireless devices (e.g., in the visualization of GUI's) and the high cost that their use implies.

In this context we are developing the ANTARCTICA[3] system to provide users of wireless devices with a new environment that fulfils their data management needs. Concerning classical data services, such as e-mail and WWW access services, ANTARCTICA goes a step further not only allowing user customization of such services, but also optimizing the performance in terms of wireless communications. Moreover, ANTARCTICA offers new services such as the Locker Rental Service, the Software Retrieval Service and the Location Dependent Queries Service that we explain briefly in section 3. These services offer new possibilities to users of wireless devices while stressing also the customization and optimization criteria.

* This work was mainly supported by *CICYT: Comisión Interministerial de Ciencia y Tecnología*, Spain [TIC2001-0660], MoviStar (a Spanish cellular phone company), the University of the Basque Country and Diputación Foral de Gipuzkoa.

[3] *Autonomous ageNT bAsed aRChitecture for cusTomized mobIle Computing Assistance.*

The architecture of ANTARCTICA is based on the use of the Client / Intercept / Server model [15, 12] and the agent technology. It incorporates modules and agents, both at the wireless devices and at intermediary elements (or proxies) situated in the fixed network. The use of agent technology gives us the means of obtaining better performance by: providing support for disconnected operations (asynchronous communications), facilitating an efficient management of data transfer interruptions and reducing the data transfer in a wireless network. Moreover, agent technology also enables to customize agents according to the task they must develop, the preferences of the user they represent and the current state and characteristics of the device and network used. While there is a small overhead in using agents, this cost is compensated by the advantages mentioned previously. On the other hand, the use of proxies increments the capabilities of wireless devices because tasks not supported by wireless devices can be performed on proxies.

In the active research areas of mobile and ubiquitous computing, several studies can be found that make extensive use of proxies and agents, separately (e.g,[13] for the case of proxies and [1] for the case of agents) or combined (e.g [4, 8, 11]. However two main features differentiate our proposal (ANTARCTICA) from the others. The first difference concerns the services provided by each intermediary element (or proxy). In our case we advocate for providing a set of services in the proxies, whereby we advocate for multi-purpose proxies, as opposed to the existing proposals that, in general, manage only one service. ANTARCTICA offers a collection of Internet data services with similar interfaces to facilitate their use. The second difference concerns the use of *majordomo* agents that are in charge of customizing the offered data services to the user needs. Moreover, ANTARCTICA supports a) *Wireless communication optimizations*: before sending data through the wireless network, the data are preprocessed, filtered or adapted to save communication costs. b) *Flexibility*: user mobility without restrictions, the specific features of user devices and the specific needs of the users concerning data services. c) *Simplicity*: the use of agents favours that technical issues become transparent to the users. d) and *Scalability*: new services can be dynamically incorporated to the system, and, services can be supported by different proxies in order to balance the global workload of the system.

The main goal of this paper is to show the performance of ANTARCTICA in scenarios that differ in the used hardware platform LAPTOP or PDA and communication protocol, GSM or GPRS. So, in the rest of this paper first, we briefly present the global architecture of the ANTARCTICA system and then the main features of the available services. Finally, some performance results and conclusions are presented.

2 Description of the ANTARCTICA System

As we mentioned in the introduction, the goal of the ANTARCTICA system is to facilitate the management of Internet data services to wireless device users.

The following elements participate in pursuing that goal: Mobile Units[4] (MUs) and the intermediary elements or proxies called *Gateway Support Nodes*[5] (GSNs). There are also agents that are situated in both elements.

2.1 MUs and GSNs

There are many different types of wireless devices, varying in capacity and size; however, a number of restrictions apply to them. Such restrictions include their limited energy supply, their relatively small capacity of memory and disk (e.g., palmtop), the constraints of the Operative System that they use (e.g., Palm) and the low quality of the wireless media that they use to communicate. In addition, they are less reliable than their stationary counterparts, since they can be more easily stolen, lost or accidentally damaged.

The GSN is the intermediary element in the communication between the mobile units under its coverage and any other host in the network, MUs or fixed hosts (FH). The pair MU-GSN allows the MU to behave like a fixed computer for the rest of the network. The GSN controls the mobile units under its coverage (their identification, profiles, groups in which they can be included, etc.).

The GSNs can be situated in different locations but always in the fixed network. Furthermore, different execution contexts called *places* are defined in a GSN. Each place contains the agents and offers the functions corresponding to each service provided by the ANTARCTICA system.

2.2 Agents

Many different definitions of what an agent is and what it is not have been proposed. In this paper we use one of the most widely accepted definitions, the one given by the *Mobile Agent System Interoperability Facilities Specification* (MASIF)[5], from which we also borrow the definitions for other basic concepts of the agent technology used in this paper. In general, an *agent* is a computer program that acts autonomously on behalf of a person or organization. An *agent system* is a platform that can create, interpret, execute, transfer and terminate agents. A *place* is a context, within an agent system, where an agent can execute (e.g., Software Place) Places can provide several functions such as the access control. Mobile agents travel among places, where the source place and the destination place can reside in the same agent system, or in different agent systems.

Different agents participate in the ANTARCTICA system. Although each type of agent is a specialist in a distinct task, they share some basic functions, so we have designed and implemented the following basic modules from which the agents in the ANTARCTICA system are built: *Communication Module*, which

[4] We use the terms *Mobile Unit* and *wireless device* as synonymous.

[5] The Gateway Support Node name is borrowed from the General Packet Radio Service (GPRS).

takes charge of the communications among agents; *Mobility Module*, that implements the agent mobility policy, controlling when and where to move; *Creation and Interaction Module*, which creates agents and manages them; and, *Specific Knowledge Module*, that contains the knowledge that agents need in order to achieve their goals and maintains the consistency and persistence of that knowledge. Each kind of agent is composed of only the needed modules, which means that not all the agents have every module.

Alfred, the majordomo agent One characteristic that differentiates ANTARCTICA from other proposals is the use of majordomo agents called Alfred, which are efficient "majordomos" for wireless device users. Alfred's behaviour is similar to the majordomo stereotype, i.e., a faithful servant to its owner. Each wireless device will have its own Alfred assigned with the aim of giving the adequate services to its user. As we have already mentioned, ANTARCTICA provides wireless device users with different data services. That means that Alfred should be a specialist in all of them. However, we advocate for a different solution. Alfred creates agents —specialized in each particular service— and entrusts the tasks required by the user to those specialist agents. In order to work with those specialist agents, Alfred consults an existing catalog in ANTARCTICA that contains information about each service and the agents responsible for that service. Although this solution produces an overhead, due to the need of managing several agents instead of only one, we consider that in this way, Alfred can better attend the user, and at the same time, each specialist agent can do its job in a more efficient way. What's more, that solution favours scalability because when new services are incorporated to ANTARCTICA, only new specialist agents are developed.

In general, majordomos remain close to their owners, so, in our proposal they are situated in the MUs. However, due to different reasons such as the limited capacity of the MU or the requirements of certain services, it may be of interest that they move to the GSNs. In that case, each Alfred has the capability of creating its own clone that can move to the corresponding GSN with the aim of there executing the services required by the user. Both agents, Alfred and its clone, play the role of user's majordomo and both are called Alfred hereafter. So, there is one Alfred situated at the MU and there may exist another one situated in the GSN. The existence of two majordomos serving to a user is transparent for the user, that is, it is an implementation level solution but the user always communicates with the Alfred situated in the MU. Moreover, when the user moves and changes from being under the coverage of a GSN to being under the coverage of a different one, Alfred, situated at the GSN also moves to the new GSN, in order to keep as close as possible to the user.

3 New Services of ANTARCTICA

In this section we show briefly the main features of novel services supported by ANTARCTICA including their goals, places and agents involved.

3.1 The Software Retrieval Service (SRS)

One of the most frequent tasks of computer users is to obtain new software, in order to improve the capabilities of their computers. Nowadays a common procedure to obtain software is to visit some of the many websites that contain freeware, shareware and demos (such as Tucows [17]). However, that procedure presents problems for many users because they must: 1) know the different programs that fulfil their needs, 2) know the features of their computers, and 3) be aware of new software and/or new releases of software of interest. Previous problems become even more important when users work with mobile computers using a wireless network media. Time dedicated to look for the software, retrieving and installing it should be minimized as much as possible, in order to reduce wireless communications and power consumed.

ANTARCTICA offers a Software Retrieval Service [9], based on the use of an ontology and the agent technology, whose goal is to allow users to find, retrieve and install software in an easy and efficient way. Easy, because with the help of intelligent agents, users can browse the ontology that describes semantically the content of a set of data sources containing pieces of software, and so, select from it, the software (the service makes the location and access method of various remote software repositories transparent for the users); and efficient, because agents take care of reducing the wireless communication cost.

Concerning related work; to our knowledge, agents have not been widely used for software retrieval. In [2] they explain a mechanism to update several remote clients connected to a server taking advantage of mobile agents capability to deal with disconnections. However this work is more related to *push technology* than to services created to assist users in the task of updating the software on their computers.

Places and agents related to the Software Retrieval Service In the following we briefly enumerate the main steps, the places and the agents involved in the Software Retrieval Service:

1. The user communicates to Alfred, situated at the MU place, the need of obtaining a new piece of software (alternatively, some keywords can be provided).
2. *The Software Manager* agent, situated at the Software place of a GSN, creates and provides a *Browser* agent with a catalog of the available software, according to the needs expressed by Alfred (on behalf of the user), i.e., the Software Manager agent is capable of obtaining customized metadata about the underlying software. For this task, the Software Manager consults the software ontology, situated in a GSN (for issues concerning the software ontology see [10]). The software itself can be either stored locally on the GSN or accessible through the Web in external data sources. Thus, the GSN can have access to a great number of distinct software for different systems, with different availability, purpose etc.

3. The goal of the Browser agent is to travel to the MU and interact with the user in order to refine a catalog of software until the user finally chooses a specific piece of software. For some refinements the Browser agent could query remotely the Software Manager agent, or travel to the GSN, query the Software Manager locally, and travel back to the MU.
4. When the user finally selects a piece of software, the Browser agent remotely creates a *Salesman* agent on the Software Place. The Salesman carries the program selected by the user to her/his computer, performs any electronic commerce interaction needed (which depends on the specific piece of software), and installs the program, whenever possible.

This approach of selecting software favours wireless communication optimization because 1) users need to visit only one catalog, stored in a GSN, instead of visiting several existing web sites, and 2) the browsing process can be done locally at the MU. Furthermore, dealing with semantic descriptions of software, instead of dealing with the different categories present in the web sites, facilitates the searching task to end users.

3.2 Location Dependent Queries Service (LDQS)

The goal of this service is the processing of location-dependent queries, i.e., queries for which the answer depends on location of wireless devices (hereafter objects). One feature that differentiates this service from others that also deal with the problem of location dependent queries is that it deals with contexts where not only the user issuing the query can change her/his position, but the objects involved in the query can move as well. A sample location-dependent query is: *"to retrieve the available police units (police stations, policemen and police cars) that are within seven miles around car38 (a stolen car), and the police cars within five miles around policeCar5 (the current chaser police car)"*. Moreover, in the considered context the service deals with continuous queries i.e., queries whose answer must be updated continuously, as opposite to instantaneous queries, also called snapshot queries in [14], for which only a single answer is obtained.

Concerning related works, the closest one to our proposal, as far as we know, is the DOMINO project [19]. However we differ in the approach used to solve the problem because, as opposite to their approach, we advocate for a completely decentralized solution for storing data about moving objects on GSNs.

Places and agents related to the Location Dependent Queries Service The Location Dependent Queries Service makes use of Location-Queries Places situated in different GSNs and of the following specialist agents: *Monitor Tracker agents*, *Tracker agents* and *Updater agents*.

When the user selects the location dependent queries service, Alfred, situated at the user mobile unit, invokes a query processor module which allows the user to pose a location query. Then, the query processor sends a Monitor Tracker agent to the GSN that provides coverage[6] to the user Mobile Unit. The goal of this agent is to minimize the wireless communications with the MU. The Monitor Tracker performs three main tasks: 1) to follow the MU wherever it goes (moving from GSN to GSN), 2) to store the data requested by the user in the case of disconnection of the MU, and 3) to refresh the data presented to the user in an efficient manner. To achieve the third task, the Monitor Tracker agent creates a net of *Tracker agents*. and send them to different GSNs. Each Tracker agent tracks a concrete reference object[7] and performs three main tasks: 1) to keep close to its referenced object, 2) to detect and process the new location of its referenced object, and 3) to detect and process the new locations of target objects[8]. For this third task, the Tracker agent creates a net of *Updater agents* to cover the relevant areas around each reference object. Concretely, each Updater agent is sent to every GSN whose coverage area intersects the relevant area around the reference object. The goal of each Updater agent is to detect the location of target objects on its hosting GSN and communicate the necessary data to its Tracker agent. For more details about this service see [6].

3.3 The Locker Rental Service (LRS)

The goal of this service is to provide the users with the possibility of keeping their data in a secure and safe space called *locker* (located on a GSN). So, the use of lockers provides wireless devices users with the following advantages. Firstly, lockers alleviate the device exposure problem (wireless devices are more vulnerable and fragile than stationary devices, because they can be easily stolen, lost or damaged). Secondly, data stored in a data locker are available for the agents at the GSN, even when the wireless device is disconnected, thus providing a solution to the availability problem (wireless devices might stay disconnected for long periods of time). Thus, specific tasks are carried out at the fixed network, with data stored in a locker, instead of on the MU, in this way relieving the media problem (wireless communications are often unstable, asymmetric and expensive). Thirdly, lockers can follow user movements, so they can reside in different GSNs but always close to the current location of the user, also relieving the media problem. Finally, due to the use of the agent technology in its implementation, lockers constitute an autonomous and auto-managed storage space (see [18] for more details).

Several commercial products are appearing, such as X-Drive [16], FreeDrive [3], mySpace [7], that also offer storage space in Internet. Their fast popularization, the number of products, and the number of customers, show the interest in this kind of services. The majority of these products offer an interface to access

---

[6] By the coverage of a GSN, we mean the total coverage of its Base Stations.

[7] Objects that are the reference of location constraint; in the sample query presented previously there are two reference objects: *car38* for the first constraint, and *policeCar5* for the second.

[8] Objects that are target of location constraints; in the sample query, *police units* for the first constraint and *police cars* for the second.

the storage space through a web browser, and some of them also offer a desktop application that enable users to directly access their accounts (FreeDrive, X-Drive). Nevertheless, those products do not provide the advantages provided by our approach. That is, they offer a passive storage space that requires the direct intervention of the user, and the rented space is situated always at the same location (in the safe hosts of the company) while in our approach, direct user intervention is not compulsory, and furthermore the rented space does not have to be always at the same location.

Places and agents related to the Locker Rental Service The Locker Rental Service makes use of Locker Places situated in different GSNs and of the following specialist agents: *locker rent agents*, *locker agents* and the *guardian agent*. When Alfred, situated in the GSN, responding to a user request, needs to use the Locker Rental Service, it creates an agent called *locker rent agent* and sends it to a Locker Place. Moreover, lockers are implemented using static agents called *locker agents*. This solution allows us to manage the locker in a dynamic and flexible manner. Each locker agent is assigned to a specific user or group of users. This pair, the locker agent and the locker rent agent, constitutes the locker itself and takes care of storing the user's data, saving e-mail messages, processing results and communicating with Alfred at the GSN. The goal of the *guardian agent* is to monitor the population of agents in the Locker Place, check the authorization and authentication of incoming agents, create and dispose locker agents, maintain a register of the agents in a database, and monitor the use of the resources.

When the Mobile Unit moves to an area covered by a different GSN, Alfred situated in that GSN also moves to the new GSN, carrying some data and agents that are working for it. For that reason, when deciding to occupy a locker, the user can specify whether s/he wants: a) the locker to move following her/his movements (the locker always remains close to the user), b) the locker to remain stationary, c) the system to decide when to move the locker, or d) to move only a small part of the locker (those data more frequently used).

4 Performance Results

In this section we analyze the performance results obtained in a scenario that makes use of a classical service such as the WWW access service, that allows users to obtain pages from their mobile units, combined with a new service provided by the ANTARCTICA system: the locker service, that allows to store the obtained pages in a locker.

The experiments have been made with the goal of measuring the influence of different variables: mobile computing models, wireless networks and portable information appliances. To perform the test we used a set of web pages with a different number of images, but with the same overall size; in particular: a) a web page of 80 Kb that contains only one figure; b) a web page of 80 Kb with 8 figures; and c) a web page of 80 Kb with 16 figures. The number of figures

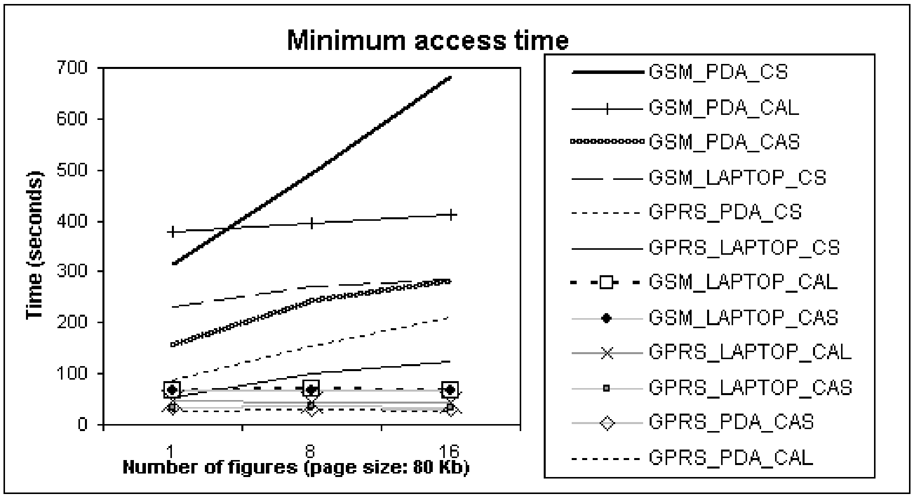

Fig. 1. Comparing all the possibilities.

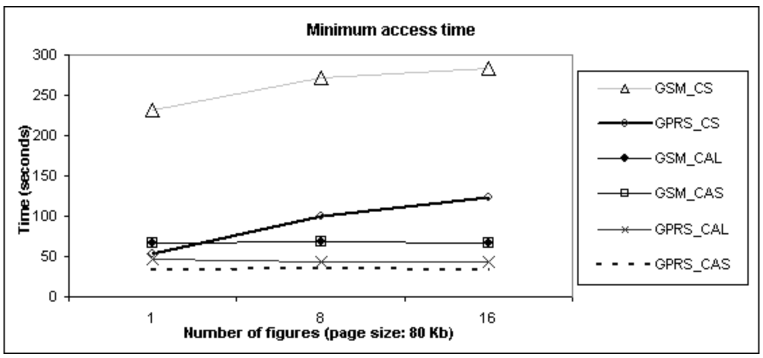

Fig. 2. Comparing wireless network and mobile computing model (using a laptop)

incrusted in a web page is important because it indicates how many interactions have to be made with the web server. We wanted to see the influence of this new variable in the performance and that is why we have decided not to change the total size of the web page. Each experiment has been repeated 10 times and the average times have been taken into account for the graphics in this section.

In the next, we detail which have been the different possibilities tried for each one of the previous variables (mobile computing models, wireless networks and portable information appliances). The **mobile computing models** are: Client/Server, Client/Agent/Server and Client/Agent/Locker.

a) Client/Server (**CS**): the MU opens a connection directly to the address where the file is located and the connection remains open until the file is downloaded.

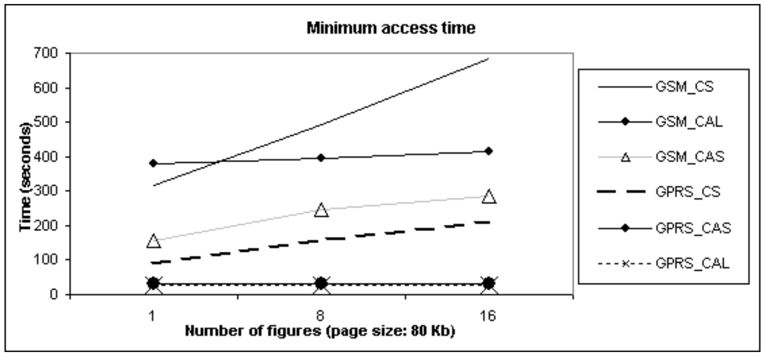

Fig. 3. Comparing wireless network and mobile computing model (using a PDA)

b) Client/Agent/Server (**CAS**): the MU sends a message (request message) to its *majordomo* agent located in the GSN, specifying the address of the file to be downloaded. Then, the MU closes the connection. The *majordomo* obtains the file and when the MU connects again, it sends the file to the MU in one message (answer message). In that way, the minimum time the MU needs to be connected to get a file by this approach is just the time needed to send and receive the request and answer messages respectively.

c) Client/Agent/Locker approach combined with the use of a locker (**CAL**): the MU sends a message (request message) to its *majordomo* agent located in the GSN, requesting a file to be obtained and stored in the locker. Then, the MU closes the connection. The *majordomo* agent obtains the file and sends it to the locker. Once connected again, the MU sends a message (read message) to the *majordomo* to read the file from the locker. Then, the *majordomo* sends the file in one message (answer message). In this way, the time the MU needs to be connected to get a file is the time needed to send the request, read and answer messages respectively. Notice that in this approach we are assuming the worst case. If the majordomo detects that the MU is connected again when having the file stored in the locker, it can take the decision of sending the file to the MU, without waiting for a specific request from the user. In this case, the time associated with the read message is eliminated. Nevertheless, the CAL approach provides the user with a persistent storage space available anywhere and at anytime, what with the CAS approach is not provided.

Two different **wireless networks** have been used: the Global System for Mobile Communications (GSM) and the General Packet Radio Service (GPRS). GSM is a connection-oriented digital wireless standard with a maximum data transfer rate of 9.6 Kbps; and GPRS is a wireless network that allows to send and receive data packets over the GSM digital wireless network. GPRS is a packet-switched network with a much higher bandwidth than GSM (about 56Kbps in our case).

And, finally, with respect to the **portable information appliance**, we have used a laptop computer (a Pentium 200MHz with 64Mb of RAM and 2Gb of HD and Windows 98) and a PDA (an HP Jornada 548 with 133MHz 32-bit processor, 32Mb of RAM, a IrDA infrared port for the communication with the mobile phone and Windows CE).

In all the experiments we were interested in calculating the *minimum* time in which the portable information appliance *had to be connected* to the wireless network when retrieving the web page. That makes much more sense when the wireless network used is GSM because you have to pay for the time in which you are connected (in GPRS you pay for the amount of data transmitted and not for the connection time). However, it is good for both of them because if connection time is less so will be the consume of the mobile phone battery that is also important.

In figure 1 it can be seen that, in general, experiments using GSM take longer than experiments using GPRS, although there are cases where using GPRS are worse than using GSM. In order to help to understand the graphics, the combinations in the legend are classified from the worst case (on the top) to the best case (on the bottom) for the case of 16 figures.

Finally, we have encountered (see figure 2) that, when using a laptop, the use of GPRS with the CS is worse than the use of GSM combined with CAL or CAS. That is, having a much faster connection than GSM does not permit us to forget about choosing a better mobile computing model like CAS or CAL. However, we have not encountered the same property for the PDA case (see figure 3). The reason for this may be the slowest connection between the PDA and the mobile phone (via the IrDA infrared port).

5 Conclusions

The number of wireless device users is growing continuously and so it is also increasing their requirements of data services. In this paper, we have shown three new services for the new mobile computing framework that help users to achieve, in an efficient way, some of their data management needs. Moreover, we report on some performance results obtained that confirm that 1) the agent technology allows to obtain a better performance than the classical CS approach and 2) the restrictions of PDAs make that the performance is worse than the obtained for laptops.

Acknowledgments: We would like to thank Gustavo Diego for helping us in performing the experiments.

References

1. H. Chen, A. Joshi, T. Finin, and D. Chakraborty. Dynamic service discovery for mobile computing: Intelligent agents meet Jini in the aether. *To appear in Journal on Cluster Computing. Special issue on Advances in Distributed and Mobile Systems and Communications.*

2. D. Chess and B. Grosof. Itinerant agents for mobile computing. Technical report, IBM Research Report RC 20010, IBM., 1995.
3. FreeDrive. http://www.freedrive.com/.
4. R. Gray, D. Rus, and D. Kotz. Agent TCL: Targeting the needs of mobile computers. *IEEE Internet Computing*, 1:58–68, August 1997.
5. Object Management Group. Mobile Agent System Interoperability Facilities Specification, November 1997. http://www.camb.opengroup.org/RI/MAF/.
6. S. Ilarri, E. Mena, and A. Illarramendi. A system based on mobile agents for tracking objects in a location-dependent query processing environment. In *Twelfth International Workshop on Database and Expert Systems Applications (DEXA'2001), Fourth International Workshop Mobility in Databases and Distributed Systems (MDSS'2001), Munich (Germany)*, pages 577–581. IEEE Computer Society, ISBN 0-7695-1230-5, September 2001.
7. YourZ.com Inc. http://www.myspace.com/.
8. E. Kovacs, K. Röhrle, and M. Reich. Mobile agents OnTheMove – integrating an agent system into the mobile middleware. In *Acts Mobile Summit. Rhodos, Greece.*, June 1998.
9. E. Mena, A. Illarramendi, and A. Goñi. A Software Retrieval Service based on Knowledge-driven Agents. In *Fith IFCIS International Conference on Cooperative Information Systems (CoopIS'2000), Springer series of Lecture Notes in Computer Science (LNCS), Eliat (Israel)*, September 2000.
10. E. Mena, A. Illarramendi, and A. Goñi. Automatic Ontology Construction for a Multiagent-based Software Gathering Service. In *proceedings of the Fourth International ICMAS'2000 Workshop on Cooperative Information Agents (CIA'2000), Springer series of Lecture Notes on Artificial Intelligence (LNAI), Boston (USA)*, July 2000.
11. S. Papastavrou, G. Samaras, and E. Pitoura. Mobile agents for WWW distributed database access. *IEEE Transactions on Knowledge and Data Engineering*, 12(5):802–820, 2000.
12. E. Pitoura and G. Samaras. *Data Management for Mobile Computing*. Kluwer Academic Publishers, 1998.
13. S. R. Ponnekanti, B. Lee, A. Fox, P. Hanrahan, and T.Winograd. ICrafter: A service framework for ubiquitous computing environments. In *Proceedings of UBICOMP 2001. Atlanta, Georgia, USA.*, 2001.
14. K. Porkaew. *Database Support for Similarity Retrieval and Querying Mobile Objects*. PhD thesis, University of Illinois at Urbana-Champaign, 2000.
15. G. Samaras and A. Pitsillides. Client/intercept: a computational model for wireless environments. In *Proceedings of the 4th International Conference on Telecommunications (ICT'97)*, April 1997.
16. X-Drive Technologies. http://www.xdrive.com/.
17. Tucows.Com Inc., 1999. http://www.tucows.com.
18. Y. Villate, A. Illarramendi, and E. Pitoura. Data Lockers: Mobile-Agent Based Middleware for the Security and Availability of Roaming Users Data. In *Fifth IFCIS International Conference on Cooperative Information Systems (CoopIS'2000), Springer series of Lecture Notes in Computer Science (LNCS), Eliat (Israel)*, September 2000.
19. O. Wolfson, A. Prasad Sistla, S. Chamberlain, and Y. Yesha. Updating and querying databases that track mobile units. *invited paper, special issue of the Distributed and Parallel Databases Journal on Mobile Data Management and Applications*, 7(3):257–287, 1999.