ANTARCTICA: A Multiagent System for Internet Data Services in a Wireless Computing Framework*

A. Goñi[†] A. Illarramendi[†] E. Mena[‡]
Y. Villate[‡] J. Rodríguez[†]

[†] *LSI depart., UPV. San Sebastián. Spain. http://siul02.si.ehu.es/~jirgbdat*
[‡] *IIS depart., Univ. of Zaragoza. Spain. http://www.cps.unizar.es/~mena*

Abstract. 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 inestability 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 paper, we present the general features of ANTARCTICA, a system which provides users of wireless devices with a new environment to achieve their data management needs. The ANTARCTICA system is based on the use of proxies and agent technology and presents the following main features: 1) *multiplicity*, managing multiple data services, 2) *scalability*, allowing one to incorporate new services, 3) *efficiency*, optimizing wireless communications, 4) *flexibility*, supporting the mobility of users without restrictions, and 5) *simplicity*, making technical issues transparent to the users. We also report on the implementation of some services supported by ANTARCTICA and present experimental results associated to them.
Keywords: mobile computing, management of data services, agent technology, Client/Intercept/Server model

1 Introduction

The sheer rapidity of the spread of wireless devices (phones, laptops, palmtops, etc.) and Internet technologies are favouring a new telecommunications revolution. The ability to access information on demand from anywhere and at anytime gives competitive advantage to individuals. Many consultancies predict that the number of wireless data users will increase considerably over the next years. However, although future wireless devices may not be as resource limited as the existing ones, users of wireless devices working with Internet data services will

* This work was mainly supported by *CICYT: Comisión Interministerial de Ciencia y Tecnología*, Spain [TIC97-0962], MoviStar (a spanish cellular phone company), the University of the Basque Country and Diputación Foral de Gipuzkoa.

still face many problems related to the nature of the devices and the communication media used.

To overcome those problems, we are developping the ANTARCTICA[1] 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 and the Software Retrieval Service that we explain in sections 3 and 4 respectively. These services offer new possibilities to users of wireless devices while stressing also the customization and optimization criteria.

The architecture of ANTARCTICA is based on the use of the Client/Intercept/Server model [SP97,PS98] 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 literature, the use of proxies and agents has been supported widely (we point out [FGCB98] related to proxy-adaptation philosophy and [GKP01] concerning mobile agent versus Client/Server performance). Particularly, in the active research areas of mobile and ubiquitous computing, several studies can be found that make extensive use of proxies and agents, separately or combined. Among those that use proxies, we point out [PLF$^+$01,Jos00,ZD97,PEBI00]. All of them agree on the idea of placing processes in the network infrastructure, rather than inserting them into end servers, with the goal of optimizing operating costs and favouring asynchronous client/server interactions. Concerning works that use agents, we highlight [CG95,HS97,SM98,CJFC]. The main idea behind them is that wireless devices send agents to the fixed network, where they work autonomously to carry out their tasks, and once they finish, they return to the wireless devices. Finally, among the works that, as in our case, use agent and proxies we mention [GRK97,KRR98,PSP00]. In [GRK97] they describe an agent system called Agent TCL and propose to use certain hosts (called docks), distributed in the fixed network where agents that cannot return to the corresponding wireless devices (because they are disconnected) remain temporarily. In [KRR98] the OnTheMove project is presented, the goal of which is to investigate

[1] *Autonomous ageNT bAsed aRChitecture for cusTomized mobIle Computing Assistance.* A preliminary version appears in [VGGI98].

a mobile middleware that supports applications for mobile users. In [PSP00] a statistical analysis appears that demonstrates the interest of using mobile agents for web database access. This last group of works is the closest to ours. 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. 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 provides two Internet data services not provided by others, the Locker Rental Service and the Software Retrieval Service for wireless device users.

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. For each service, we introduce its goal, the advantages that it provides to its users and some experimental results. Finally, some 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. In figure 1 we observe the elements that participate in pursuing that goal: Mobile Units[2] (MUs) and the intermediary elements or proxies called GSNs. Agents and Places are situated in both elements.

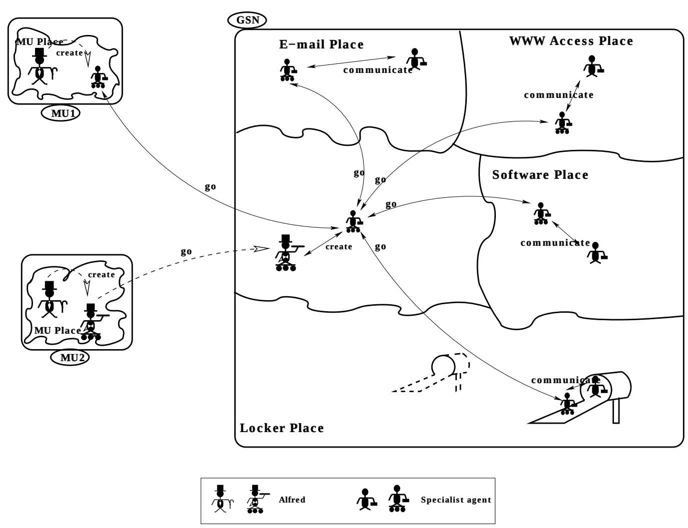


Fig. 1. Elements involved in the ANTARCTICA system.

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

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) 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 depending on the considered scenario. In a scenario in which a cellular phone company offers the data services, the GSNs will belong to that company and can be situated as shown in figure 2.a (notice that the figure represents the widely accepted architecture for mobile computing [PS98,HHC^{+}99] extended with our proposal of supporting GSNs). GSNs are always situated in a fixed network. The number of GSNs in the network will depend on the number of users. Typically, a GSN will cover an area controlled by several Base Stations (BS). However, in areas with high concentration of users, there would be more GSNs. In such cases the GSN could even reside in the Base Station (BS). In a different scenario, they would be located on a computer located in the intranet of a private organization, in charge of monitoring the access to the corporate network (see figure 2.b).

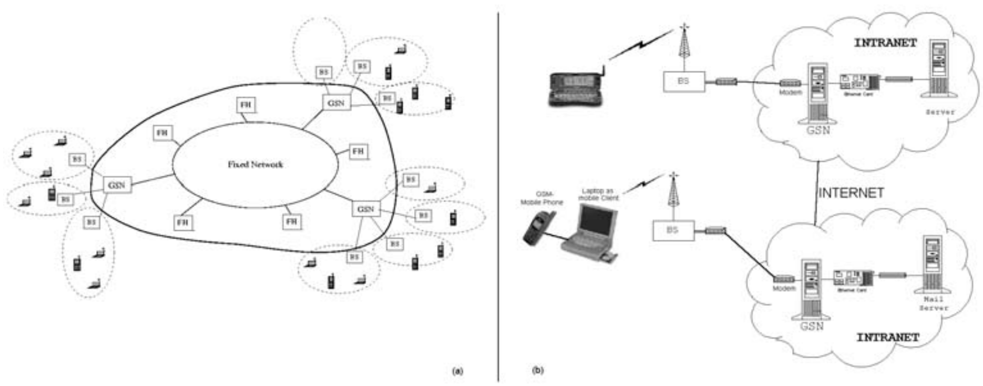


Fig. 2. (a) GSN owned by a cellular company, and (b) GSN owned by a particular company.

Furthermore, different execution contexts called *places* are defined in a GSN (see figure 1) 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)[Gro97], 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., Locker Place, Software Place in figure 1). 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.

Alfred, the majordomo agent, and other specialist agents of the system

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. 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 developped.

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.

As may be noticed in figure 1, 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 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.

In the following sections we show the main features of the four services currently supported by ANTARCTICA including their goals, places and agents involved, as well as certain experimental results.

3 The Locker Rental Service

This service offers users the possibility of keeping their data in a secure and safe space called *locker* (located on a GSN). In summary, 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 [VIP00] for more details).

Several commercial products are appearing, such as X-Drive [Tec], FreeDrive [Fre], mySpace [Inc], that also offer storage space in Internet. Their fast popularization, the number of products, and the number of customers, demostrate the interest in this kind of services. The majority of these products offer an interface to access 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 necessary, and furthermore the rented space does not have to be always at the same location.

Places and agents related to the Locker Rental Service

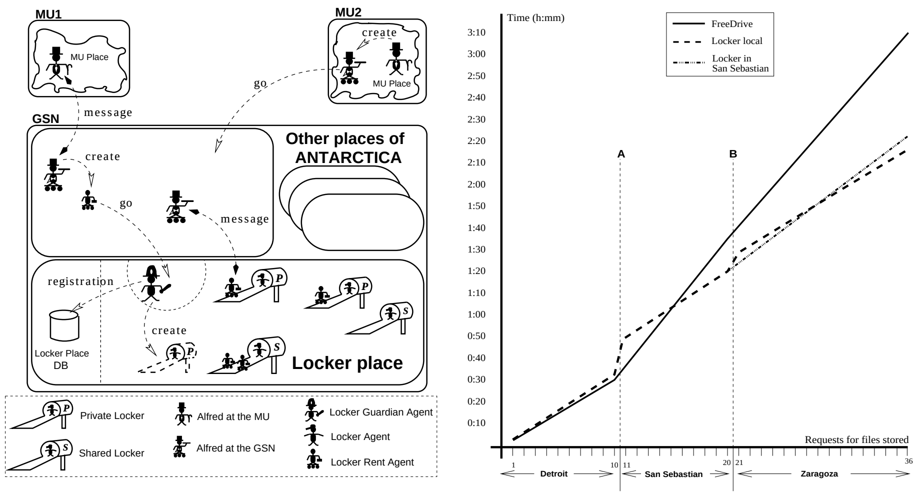
The Locker Rental Service makes use of Locker Places situated in different GSNs and of the following specialist agents: *locker rent agents* (one per locker, user of the locker and locker place), *locker agents* (one per locker and locker place) and the *guardian agent* (one per locker place) (see Figure 3.a). 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).

Experimental Results

In figure 3.b we present a scenario in which we compare the FreeDrive approach and the Locker Rental Service of the ANTARCTICA system. We suppose that a user, using a wireless media, accesses some files[3] from Detroit, then he travels to San Sebastian (Spain), requests more files, next he travels to Zaragoza (300 km far from San Sebastian) and continues accessing files. We explain the three different situations that can arise as shown in figure 3.b: 1) using FreeDrive, 2) using the Locker Rental Service with the option of having the locker always following the user, and 3) using the Locker Rental Service leaving the system choose when to move the locker or not. 1) *Using FreeDrive.* The user accesses his files stored in the safe computers of FreeDrive which are located in Detroit. We consider that it takes 192 seconds to download each file stored in FreeDrive. After some time (denoted as A in figure 3.b), the user leaves Detroit and travels to San Sebastian. Once there the user connects his MU again, in order to continue working. Now, the time it takes to access a 150 K file stored at FreeDrive is

[3] We assume that the size of each file accessed is 150 K.



(a) (b)

Fig. 3. (a) Arquitecture and (b) experimental results for the LRS

384 seconds, as files are located in Detroit. After some time (denoted as B in figure 3.b), the user travels to Zaragoza, connects his MU and continue accessing other files. We have considered that the time needed to access a file from Zaragoza is similar to the time needed from San Sebastian (the bottleneck is the connection between Spain and the USA). 2) *Using the Locker Rental Service*. First, the user accesses his files from Detroit. We suppose that the locker is currently located at a GSN in Detroit. However, we assume that it needs 200 seconds to download each file (8 seconds more than using FreeDrive), highly penalizing the ANTARCTICA approach because of the messages interchanged by the agents. When the user travels to San Sebastian, the Locker Rental Service decides to move the locker to a GSN in that city. We have assumed a locker size of 40 M. However, as the transfer is performed from a GSN in Detroit to a GSN in San Sebastian (GSN's are connected by a wired network), the locker movement only takes 13 minutes and 39 seconds. Now the access times to files of the user from San Sebastian to his locker remain the same as in Detroit (200 seconds), as the locker is as close to the user as possible. Notice that locker is not moved at the time the user makes the first request but the locker is moved at the time the system detects the user has switch on his device at a new location. That means the locker moves on its own, without affecting the user work with the system, unless the user requests something from the locker exactly while it is moving, in which case he will have to wait until the locker ends the moving. This worst case is the one supposed in figure 3. When the user travels to Zaragoza the lockers moves from San Sebastian to Zaragoza. But now the transfer of 40 M is faster (we have tested that a transfer is about two times faster between Zaragoza and San

Sebastian, than between San Sebastian and Detroit), so it only takes 6 minutes and 49 seconds. File accesses from Zaragoza remain the same (200 seconds) and the locker is still as close to the user as possible. 3) *Using the Locker Rental Service but the system chooses to leave the user locker at San Sebastian.* Then, when the user travels to Zaragoza, the locker is not moved, but now the access to the file stored in the locker takes 256 seconds, 56 seconds more, due to the need of accessing from Zaragoza to the locker in San Sebastian.

Notice in figure 3.b) that if the user accesses more than 27 files, the locker strategy "follow the user wherever he goes" is the best.

4 The Software Retrieval Service

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 [Tuc99]). 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[4] 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 [MIG00a], based on the use of an ontology and the agent technology, that allows 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 [CG95] 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.

4.1 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 (see Figure 4.a):

[4] For this service, we consider that the wireless devices are laptops, that is, computers capable of executing agents and where general-purpose software can be installed.

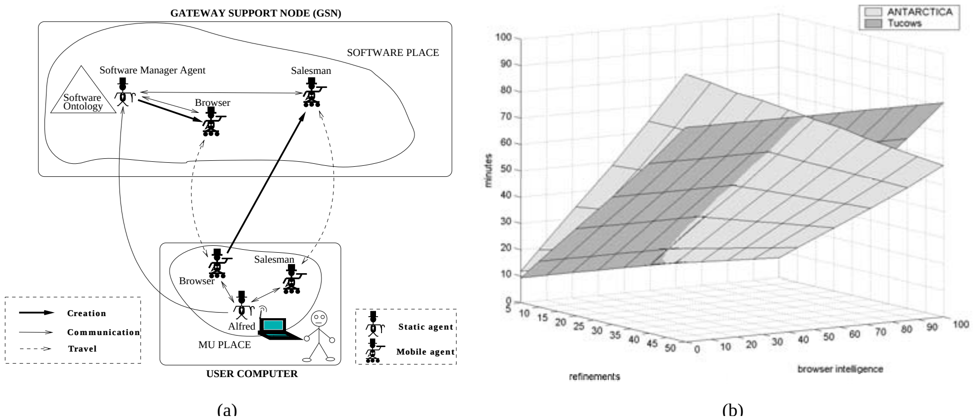


Fig. 4. (a) Arquitecture and (b) experimental results for the SRS (net speed 1 K/sec, webpage/catalog 50 K, user delay 60 sec)

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, 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 the GSN (for issues concerning the software ontology see [MIG00b]). 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.

4.2 Experimental Results

Up to now, the traditional way to obtain new software is by navigating HTML pages in public or pay-per-download software repositories (e.g., Tucows [Tuc99]). These repositories classify the different software in categories. The user selects a category, browses the information, clicks on a link which involves a remote access, browses the information obtained, clicks again on another link and repeats this process until s/he requests a piece of software. Our proposal is different, so we were interested in knowing when each one of both possibilities is better: HTML navigation (which corresponds to a traditional Client/Server approach) or our proposed Software Retrieval Service.

The key to the success of our approach, when compared to the Client/Server approach, is the number of refinements that the Browser agent is able to perform without external help, which also depends directly on the size of the initial catalog as well as on the "intelligence" of the Browser agent.

In Figure 4.b we can observe some performance tests obtained by using an analytic model of the Software Retrieval Service, based on UML and Petri nets, developped in [MCM00]. Data have been obtained for a scenario with a net speed of 1 K/sec (a wireless network), a webpage/catalog size of 50 K and a user delay between refinements of 60 sec (naive users). In order to study the possible bottlenecks, we have taken into account the following parameters: 1) The number of refinements requested by the user, 2) The probability that the Browser agent does not need new information under a single user request (hereafter, the Browser agent "intelligence"), and 3) the time needed to retrieve the piece of software. The "Browser intelligence" axis measures the percentage of user refinements that the Browser agent is able to attend to by itself, without consulting the Software Manager agent. Notice that, in that scenario, our approach is better as it reaches an intelligence greater than 50%, which is very easy to achieve as the Browser agent manages a catalog of software customized for each user. Also, both approaches are very similar for few refinements (for expert users, for example, who need just a few clicks to find the software that they are looking for).

We would like to stress that many other situations which would require a big effort in implementation can be obtained easily with this formal model of the Software Retrieval Service, annotated with empirical data obtained from much simpler experiments (e.g., size of agents, messages, etc).

5 Traditional services: WWW access and e-mail

The WWW access service allows users to obtain web pages from their wireless devices, and also to navigate through the Internet, optimizing wireless communications and connection time. The e-mail service permits users to send and receive e-mails anywhere and at anytime in a flexible and efficient way. Moreover, it also allows users to get a customized management of their e-mails even without their direct intervention.

Nowadays, there are so many proposals for the services considered in this subsection that it results almost impossible to contribute with original ideas. For that reason, instead of presenting the particular features of the WWW access service and e-mail service supported by ANTARCTICA, we show in this section two tests which we carried out for the WWW access, in order to compare its behaviour, that follows a Client/Agent/Server (C/A/S) approach, with respect to the traditional Client/Server (C/S) approach.

5.1 Test 1

For this test, we used two web pages[5]. The first page, located in a national server (situated 9 hops away from the MU and 9 hops away from the GSN) contained 12 pictures and 91.6 K of data. The second page, located in an international server contained 11 pictures and 53.2 K of data. The test consisted of measuring the time needed to access the pages depending on the type of communication supported by the device, i.e., GSM, POTS (Plain Old Telephone System), and Ethernet.

We did two types of analysis with the obtained times: a descriptive statistical analysis and a statistical test. For both we used SPSS [fW] to analyze the data sets. According to the descriptive statistical analysis, we were interested in the mean and the standard deviation of the data sets (see figure 5 for the page located in an international server). Observing the mean we concluded that the access using our approach was more efficient when low bandwidth communication media were used, such as POTS or GSM. However, for Ethernet, where the bandwidth between the MU and the GSN is similar to the one used between the GSN and the WWW server, the results obtained by both approaches were comparable, being slightly better the results obtained by the traditional C/S approach. Another important outcome was derived from the analysis of the standard deviation of the data sets. That analysis showed that when low bandwidth media were used, and in particular GSM, the standard deviation of our approach was lower than the obtained using a C/S approach. This behaviour was attributed to the fact that the C/S approach continuously used the communication media, and so it was influenced by its random behaviour, while dealing with agents this disturbing fluctuation was avoided since communications between the MU and the GSN were reduced.

For the second type of analysis, that is, a statistical analysis of the mean for each one of the used approaches (ours and C/S) we used the t-test[6] for independent samples. The null hypothesis of the t-test said that: "*the mean time for obtaining a Web page using our approach was equal to that obtained using the C/S approach* (H_0: $\mu_1 = \mu_2$, H_1: $\mu_1 \neq \mu_2$)". Looking at the tables A1, A2 and

[5] No HTTP proxy was used.

[6] t-hypothesis test to compare two means under normality assumption.

Mean in ms.:

	GSM	POTS 14.4	ETHERNET
CAS Mean	104124,8	60372,8	73932,9
CS Mean	239926,6	152621,2	62601,7

Standard Deviation in ms.:

	GSM	POTS 14.4	ETHERNET
CAS SD	39396,3	27047,2	42532
CS SD	112758,8	28546,7	40513,4

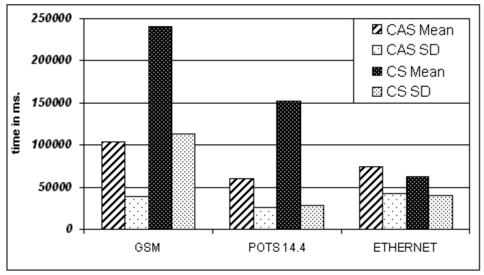


Fig. 5. Access to a WWW page located in an international server.

A3 in the Appendix A (obtained for the page situated in the national server), we concluded[7] that the observed differences between the mean time needed for obtaining a Web page were statistically significant for our approach only when the GSM and the POTS communication media were used. In both cases, GSM and POTS, the p-value obtained was $p<0.001$ for a t-test with *unequal* variances. Therefore, the t-test proved that the time required for accessing a Web page using our approach was always less than the time required when the C/S approach was used, so the null hypothesis was rejected. However, when Ethernet was used the null hypothesis was accepted, so the means for both approaches cannot be considered distinct.

5.2 Test 2

The goal of this test was to analyze the behaviour of both approaches (C/S and C/A/S) when accessing to different web pages situated in the same server. Each page had a different amount of data and contained a different number of pictures. More precisely, 20 different web pages were used with data varying from 20 K to 160 K and the number of picture files varying from 1 to 16. The data sets for each test were obtained from more than 40 accesses distributed at different daytimes. From the analysis of the obtained results we concluded that our approach behaviour was better than the C/S approach when the number of interactions between the client and the server increased due to the number of pictures contained in the page. The reason for this behaviour was that the protocol needed to fulfil these interactions was moved to the intermediary element that had a higher bandwidth than the 9.600 bps. used in GSM connection. Another conclusion was that our approach also worked better when the amount of data required by the user increased.

[7] A similar conclusion was reached in the case of the international page.

6 Conclusions and future work

In this paper we have presented ANTARCTICA, our proposal to help wireless device users to achieve their data management needs. ANTARCTICA takes into account the restrictions that, nowadays, the mobile computing framework presents and tries to overcome them by using the agent technology and intermediary elements situated in the fixed network. In summary, ANTARCTICA offers the following advantages: 1) *Multiple services.* As oppossed 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. 2) *Wireless communication optimizations.* Before sending data through the wireless network, the data are preprocessed, filtered or adapted to save communication costs. 3) *Flexibility.* ANTARCTICA supports: user mobility without restrictions, the specific features of user devices and the specific needs of the users concerning data services. 4) *Simplicity.* The use of agents favours that technical issues become transparent to the users. 5) *Scalability.* New services can be dinamically incorporated to the system. Moreover, services can be supported by different proxies in order to balance the global workload of the system. 6) *Modularity.* The modular design of ANTARCTICA permits one to reuse modules for different services.

As future work, we plan to incorporate new Internet data services to ANTARCTICA, that support e-commerce, and to adapt the system to the features of the UMTS (Universal Mobile Telecommunications System) wireless communication protocol.

References

[CG95] D. Chess and B. Grosof. Itinerant agents for mobile computing. Technical report, IBM Research Report RC 20010, IBM., 1995.

[CJFC] 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.*

[FGCB98] A. Fox, S. D. Gribble, Y. Chawathe, and E. A. Brewer. Adapting to network and client variation using active proxies: Lessons and perspectives. *IEEE Personal Communications*, August 1998.

[Fre] FreeDrive. http://www.freedrive.com/.

[fW] SPSS for Windows. Rel. 8.0.0.0 1997. Chicago. SPSS Inc.

[GKP01] R. S. Gray, D. Kotz, and R. S. Peterson. Mobile-agent versus client/server performace: Scalability in an information-retrieval task. Technical report, Dartmouth College Computer Science Technical Report TR2001-386, January 2001.

[GRK97] R. Gray, D. Rus, and D. Kotz. Agent TCL: Targeting the needs of mobile computers. *IEEE Internet Computing*, 1:58–68, August 1997.

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

[HHC+99] A. Helal, B. Haskell, J.L. Carter, R. Brice, D. Woelk, and M. Rusinkiewicz. *Any Time, Anywher Computing. Mobile Computing Concepts and Technology*. Kluwer Academic Publishers, 1999.

[HS97] L. Hurst and F. Somers. Mobile agents - smart messages. In *Proceedings of the First International Workshop on Mobile Agents (MA97)*, 1997.

[Inc] YourZ.com Inc. http://www.myspace.com/.

[Jos00] A. Joshi. On proxy agents, mobility, and web access. *Mobile Networks and Applications*, 5:233–241, 2000.

[KRR98] 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.

[MCM00] J. Merseguer, J. Campos, and E. Mena. Performance evaluation for the design of agent-based systems: A Petri net approach. In Mauro Pezzé and Sol M. Shatz, editors, *Proceedings of the Workshop on Software Engineering and Petri Nets, within the 21st International Conference on Application and Theory of Petri Nets*, pages 1–20, Aarhus, Denmark, June 2000. University of Aarhus.

[MIG00a] 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.

[MIG00b] 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.

[PEBI00] S. H. Phatak, V. Esakki, B. R. Badrinath, and L. Iftode. Web&: An architecture for non-interactive web. Technical report, Rutgers University, Department of Computer Science, Technical report DCS-TR-405, 2000.

[PLF+01] 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.

[PS98] E. Pitoura and G. Samaras. *Data Management for Mobile Computing*. Kluwer Academic Publishers, 1998.

[PSP00] 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.

[SM98] A. Sahai and C. Morin. Mobile agents for enabling mobile user aware applications. In *Proceedings of the Second International Conference ACM Autonomous Agents (Agents 98)*, 1998.

[SP97] 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.

[Tec] X-Drive Technologies. http://www.xdrive.com/.

[Tuc99] Tucows.Com Inc., 1999. http://www.tucows.com.

[VGGI98] Y. Villate, D. Gil, A. Goñi, and A. Illarramendi. Mobile agents for providing mobile computers with data services. In *Proceedings of the Ninth IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 98)*, 1998.

[VIP00] Y. Villate, A. Illarramendi, and E. Pitoura. Data Lockers: Mobile-Agent Based Middleware for the Security and Availability of Roaming Users Data. In *IFCIS International Conference on Cooperative Information Systems (CoopIS'2000), Springer series of Lecture Notes in Computer Science (LNCS), Eliat (Israel)*, September 2000.

[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.

A APPENDIX

A.1 National Server using 9.600bps GSM

Variable	Number of Cases	Mean	SD	SE of Mean
CS	137	200415.1	85310.1	7288.5
CAS	188	132980.8	28594.4	2085.4

Mean Difference = 67434.3
Levene´s Test for Equality of Variances F=23,77 p<0,001
t-test for Equality of Means 95%

Variances	t-value	df	2-Tail Sig.	SE of Diff	CI for Diff
Equal	10.09	323	.000	6681.3	(54289.8; 80578.7)
Unequal	8.90	158.4	.000	7581.0	(52461.4; 82407.2)

A.2 National Server using 14.400bps POTS

Variable	Number of Cases	Mean	SD	SE of Mean
CS	65	148888.7	80666.8	10005.4
CAS	65	78573.5	11851.7	1470.0

Mean Difference = 70315.2
Levene´s Test for Equality of Variances F=50.99 p<0,001
t-test for Equality of Means 95%

Variances	t-value	df	2-Tail Sig.	SE of Diff	CI for Diff
Equal	6.95	128	.000	10112.9	(50305.1; 90325.3)
Unequal	6.95	66.76	.000	7581.0	(50128.4; 90501.9)

A.3 National Server using 10Mbps Ethernet

Variable	Number of Cases	Mean	SD	SE of Mean
CS	144	57540.7	45975.9	3831.3
CAS	117	60287.6	55070.1	5091.2

Mean Difference = -2746.8
Levene´s Test for Equality of Variances F=0.016 p=0,901
t-test for Equality of Means 95%

Variances	t-value	df	2-Tail Sig.	SE of Diff	CI for Diff
Equal	-0.44	259	.661	6254.7	(-15063.4; 9569.7)
Unequal	-0.43	225.83	.667	6371.7	(-15302.6; 9808.9)