Extending the Data Services of Mobile Computers by External Data Lockers*

Yolanda Villate[†]
Basque Country University, CS Dept.
jibvipey@si.ehu.es

Evaggelia Pitoura
University of Ioannina, CS Dept.
pitoura@cs.uoi.gr

Arantza Illarramendi
Basque Country University, CS Dept.
jipileca@si.ehu.es

Ahmed K. Elmagarmid
Purdue University, CS Dept.
ake@cs.purdue.edu

Abstract

Although mobile computers are becoming more and more powerful, the intrinsic features of wireless communications - poor quality, limited bandwidth, continuous disconnections, expensive communications - still limit the performance delivered to mobile users. In this paper, we present the Locker Rent Service that allows mobile users to rent a dedicated disk space, located in the fixed network, where they can store and access their data. Besides increasing the storage capacity of mobile devices, the Locker Rent Service offers to mobile users data protection, battery power savings and various communications optimizations. The service is supported by a middleware architecture that incorporates this and other services with the general goal of increasing mobile computers capabilities and performance. The architecture is based on mobile agents and offers flexibility and adaptability with a low overhead as shown by our preliminary performance results.

1. Introduction

In the past few years, there has been a widely spread use of portable computers and wireless networks. The combination of both leads to a new technology: *mobile computing*. Mobile computing allows users to access from any place and at any time data stored in the repositories of their organizations as well as data available in a global information system through the Internet. However, the intrinsic features of wireless communications - poor quality, limited bandwidth, continuous disconnections, expensive communications - make it problematic and monetarily expensive to work continuously connected to a wireless network.

In [12], we introduced the ANTARCTICA (Autonomous ageNT bAsed aRChitecture for cusTomized mobIle Computing Assistance) system whose main goal is to alleviate the problems of mobile computing. The architecture of the system is based on the use of the *client/intercept/server* model [10] and incorporates modules and agents both at the mobile device and at an intermediary element, or proxy, situated at the fixed network called the *Gateway Support Node*[1](GSN). ANTARCTICA offers a set of data services that enhance the capabilities of the mobile unit (MU) and offers new possibilities to mobile users.

In this paper, we describe the design and implementation of a service central to ANTARCTICA: the *Rent of Lockers*. The basic idea of the *Locker Rent Service* is to provide mechanisms that allow mobile users to use storage space external to their mobile computers by renting disk space in the intermediary element or the GSN. By renting this space or locker, a user can somehow "expand" his MU to the GSN by storing data of interest in a safe and persistent way, without having to download them to the MU. The Locker Rent Service offers several advantages to the users of mobile units:

1. **Storage space:** The locker becomes an extension of the MU's disk placed at the fixed network.
2. **Data protection:** The data stored in the locker are protected against unauthorized accesses and modifications as well as any unexpected failures.
3. **Battery power saving:** The mobile client can stay disconnected for longer periods of time, while the data obtained for it are sent to its locker and stored there until it reconnects or explicitly asks for them.

*This work is supported by *CICYT: Comisión Interministerial de Ciencia y Tecnología*, Spain. [TIC97-0962]; and MOVISTAR, a Spanish Cellular Phone Company.

[†]The work of this author was supported by a grant of the University of the Basque Country.

[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) for the second phase of the GSM standard. We take GSM and GPRS as cellular network model for our work.

4. **Wireless communications optimizations:** The space in the locker can be used to store data until it is possible or desirable to send them to the MU. Before sending data to the MU, the data can be preprocessed, filtered or adapted to the needs of the mobile unit and its user.

There are two lines of research related though not exactly similar to ours. The first line includes research that relies on the idea of renting resources, such as for example [3] where the main focus is on accounting CPU cycles and communications consumption made by the hosted processes. On the contrary, the main idea of our service is to rent storage. The second line exploits the use of *proxies* and software agents [7, 6, 4, 13, 5, 8, 9]. To our knowledge, our work is the first one to combine both aspects, by employing proxies and agents to provide users with storage external to their mobile computers. Finally, our work can benefit from other works, such as the CODA project [7], by implementing the ideas and techniques they propose for the maintenance of data consistency among copies stored in the MU cache, the GSN and the servers.

The remainder of this paper is structured as follows. In Section 2, we describe the architecture of the system in detail, along with specific features such as the types of lockers, mobility and security. In Section 3, we present our prototype implementation along with preliminary performance results. Finally in Section 4, we present our conclusions.

2. The Locker Architecture

The locker service is offered within the ANTARCTICA system. The architecture of ANTARCTICA is based on mobile agents. Agents travel between execution environments called *places* [2], where they can execute. In particular, at each MU, there is a static agent, running on a place in the MU. This static agent isolates the user's applications from network availability and the specific characteristics of the MU and is responsible for the administration of the MU's resources. In addition, this static agent creates and launches a single agent from the mobile unit to the fixed network. This agent is called the *majordomo agent*. Once launched, this agent remains in the GSN for the time period required, working on behalf of the MU even while the MU is disconnected. Each mobile computer has its own *majordomo* with the aim of providing adequate services to its owner. Communications of the MU with the exterior are surveyed by this pair of agents. Both agents work together in order to adapt and optimize the communications and use of the wireless media by the MU.

In the GSN, there is a place called *Inventory* where majordomo agents execute and can get the information they need about other GSNs, places and services in the system. Besides that, there are specialist places, one of which is the *Locker Place*, that offers the *Locker Rent Service*. This service offers disk space for renting, so it has mechanisms to administer and monitor the available space, to assign spaces, and to register data necessary for billing for the service. When the *majordomo* needs to use the *Locker Rent Service*, it creates an agent called *locker rent agent* and sends it to a *Locker Place*. Although it would be possible for the majordomo agent to use directly the *Locker Rent Service*, we chose to create a specialist agent because it would be difficult for the majordomo to use efficiently all the different services offered by the ANTARCTICA system heeding at the same time the different user requirements. Figure 1 shows the elements involved with the *Locker Rent Service*.

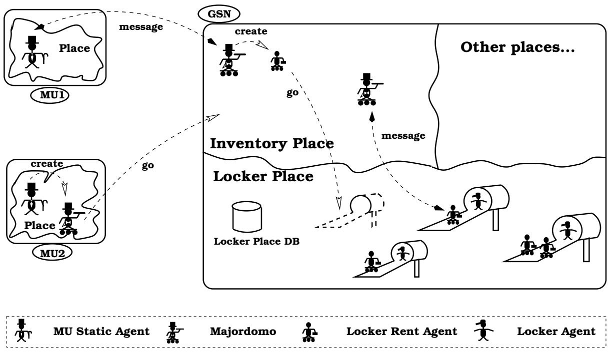

Figure 1. Elements in Locker Rent Service.

2.1. The Pair of Locker Agents

Lockers are implemented using a new kind of agents that we call *locker agents*. *Locker agents* are created with space assigned in which they can store data. This means that lockers are created as they are rented by new users, so there are no physically separated compartments to reserve. Each *locker agent* occupies disk space according to its needs and limitations. When it needs more space, the agent occupies it, and when it needs less, the agent releases space. The *locker agents* are created by the *Locker Place* and each one of them is assigned to a specific user or group of users, i.e., to their *locker rent agents*. This pair of the *locker agent* and the *locker rent agent(s)* constitutes the locker and takes care of storing the user's data, saving email messages, processing results and communicating with the MU's *majordomo* agent. The fact that both agents have to communicate with each other and interchange data incurs some overhead. However, this interaction is local as both agents reside at the same place. Furthermore, by having both kinds of agents, we can take advantage of specializing them.

2.2. Types of Lockers

Lockers belong to categories. Lockers of different categories vary in the maximum size they can reach, the services they include, their prices and the way requests for extending their maximum size are handled. In addition, the *Locker Rent Service* provides two basic kinds of lockers: private and shared lockers.

A **private locker** is related to a single user; the data stored belongs to that particular user and can be accessed or modified only by authorized agents representing the user. A **shared locker** is a locker rented by a group of users. A shared locker distinguishes between data to be used by all users in the group and data for each particular user, by managing sub-lockers for each of them. Moreover, the space in the locker that is shared by all users of the group constitutes an encounter place for these users who are able to communicate, share data, interchange messages, write messages for the rest of the group in a blackboard and collaborate.

A private locker is created when a user asks to use the service and is destroyed when the user decides to vacate the locker. A shared locker is created when a related contract is signed and persists for the whole duration of the contract. That is, in contrast to a private locker, a shared locker exists even when no member of the group is using it.

2.3. Mobility of the Locker

Private lockers are typically created in the GSN under the coverage of which the MU is located. In our system, the *majordomo* agent is located as close as possible to the MU it belongs to. This means that when a MU moves to an area covered by a different GSN, its *majordomo* agent also moves to the new GSN carrying with it its data and (some of) the agents that are working for it. When deciding to occupy a locker, the user can specify whether he wants (a) the locker to move following his movements, (b) the locker to remain stationary, or (c) the system to decide when to move the locker.

The system decides on whether to move a locker or not considering the cost-benefit of the transmission. Prior to the computation of the cost of the transmission of the locker and its contents, the *Locker Place* must negotiate to find another *Locker Place*, located closer to the new GSN of the *majordomo*, which can accept the locker being transmitted. Here the benefit is understood as the time that can be saved by having the agents interacting closer. In general, moving the locker from one GSN to another is an expensive operation. Nevertheless, the possibility of moving a locker is useful since the system can choose to move a number of lockers from one GSN to another to balance the system load, while taking care of not deteriorate the interaction between the lockers and their *majordomos* by overly incrementing the cost of their communications.

The shared lockers are associated with a group of users whose movement is in general independent. Their location is the result of a decision of the group or the representative of the group, so shared lockers do not necessarily move following their users.

2.4. Security

In the context of mobile agents, security is always a principal concern because generally agents can not trust the hosts they visit, and hosts can not trust the agents they receive. The protection of agents against malicious hosts is an open problem still without a completely satisfactory solution. The approach most commonly used is to rely on an organizational framework whose hosts can be trusted [11]. This is the case in our system, since we assume that the services offered by the GSN are provided by a trusted cellular phone company.

The security of hosts against agents can be achieved satisfactorily by using existing techniques [11]. The central issue is to prevent uncontrolled access and use of the host resources. In our system, the only agents that are created from classes unknown to the GSNs are the *majordomos*. It must be ensured that such agents represent authorized users. To achieve this, credentials are associated with the agents that correspond to the identity of their user, access level and key. Such credentials are used for their authentication and authorization, and for controlling their access to services and data by checking them against the information the system maintains about registered users.

Apart from the use of mobile agents, in the Locker Rent Service, data stored in a private locker belong to the user and are protected against unauthorized access from other users or their agents. In a shared locker, members of the group owning the locker are allowed to access the data that is specifically stored as common, while each user in the group has a private area where he can store information that nobody else has the right to access. This is achieved by not allowing the *locker agents* access any other disk space but their own. When a *locker rent agent* makes a request to a *locker agent*, the *locker agent* checks the authenticity of the *locker rent agent* and decides to which data it has access to. Finally, the use of the resources of the *Locker Place* must be controlled by the system in order to monitor the use of space, administrate the resources and avoid intrusions, as well as to implement the accounting mechanisms and policies.

3. Implementation and Performance Results

A prototype implementation of the Locker Rent Service using the platform “Aglets Workbench” [1] is under development. At the present, our prototype allows the creation of private lockers, the storage of files in the lockers, authorized access to the files, and the release of the lockers.

We have performed a number of experiments using on one hand, a portable computer equipped with a 233 MHz Pentium processor, 64 M of RAM, running Windows 98. This machine played the role of the MU. For the tests we present in this paper the MU used a 9,600 bps GSM card as communication device. On the other hand, we used a fixed computer equipped with a 400 MHz Pentium II with 128 M of RAM running Linux Red Hat 6.1. The latter machine played the role of the GSN. We have performed several experiments[2], in which we compared the time that the MU needs to be connected in order to get, for example, a file according to the following three approaches:

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.

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 can close 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.

c) Client/Agent/Server 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 can close 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 that 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.

When comparing the CAS with the CAL approach, the use of the locker in the CAL approach introduces a time overhead due to the read message, but the ability of storing data for the MU in a persistent and safe way constitutes the strong point of the CAL approach over the CAS one.

Notice that we are only interested in the MU connection time and not in the total cost. Saving connection time also means that the MU saves battery power and money. The CAS and CAL approaches make it possible to work in an asynchronous way, i.e., the MU submits a task to be performed and then disconnects or continues working on other tasks, while the submitted task is being executed in the GSN. When the task finishes, the MU receives the results.

Our experiments showed that when downloading a single file, the CS approach performs slightly better than the approaches using agents. The time the MU needs to be connected is determined by the time it takes to interchange data between the MU and the GSN. With the CS approach that time is mostly the time it takes to send the file through the wireless media. However, in the CAS and CAL approaches there are some messages interchanged between the agents situated in the MU and in the GSN. Even if these messages are short, the transmission of them through the GSM media at 9,600 bps causes the small overhead of those approaches over the CS one. However, the bigger the size of the file, the more irrelevant the overhead. Nevertheless, these results do not constitute a proof against the use of the CAS and CAL approaches. On the contrary, taking into account that the cost that influences the most the final cost is the one associated with the low speed communication media, the approaches using the intermediary agent give the opportunity to reduce the communication cost by, for example, (a) applying filters in the GSN that reduce the amount of data to be transmitted; and (b) by avoiding interactions between the MU and the GSN, i.e. the need of using the wireless media.

In the first experiment we present here, a compression filter was applied to the files before transmitting them to the MU. The filter applies a *lossless compression*. Table 1 shows the characteristics of the files used. Figure 2.a shows the average time the MU has to be connected to get the files by each of the approaches considered, when using the wireless media. These results show how even a small compression percentage is enough to make the approaches with agents better than the CS approach.

Table 1. Set of files used for test 1.

Name	File Type	Original Size	Compressed Size	Compression %	Avg. compr. time
test2.pdf	text	161.76Kb	154.72Kb	4	146 ms
test1.pdf	text	184.96Kb	154.55Kb	16	180 ms
test1.au	audio	172.14Kb	90.02Kb	48	237 ms
test1.mid	audio	153.53 Kb	8.96Kb	94	45 ms

Our second experiment shows optimizations that can be obtained by avoiding interactions between the MU and its outside world, i.e. avoiding the use of the wireless media. For this test, the user asks for a web page to be obtained, i.e. an HTML file along with the image files that are contained in the page. When using the traditional CS approach, a new connection has to be open for each of the files to be downloaded to the MU. However, when using the ap-

[2] Each experiment has been performed more than 20 times.

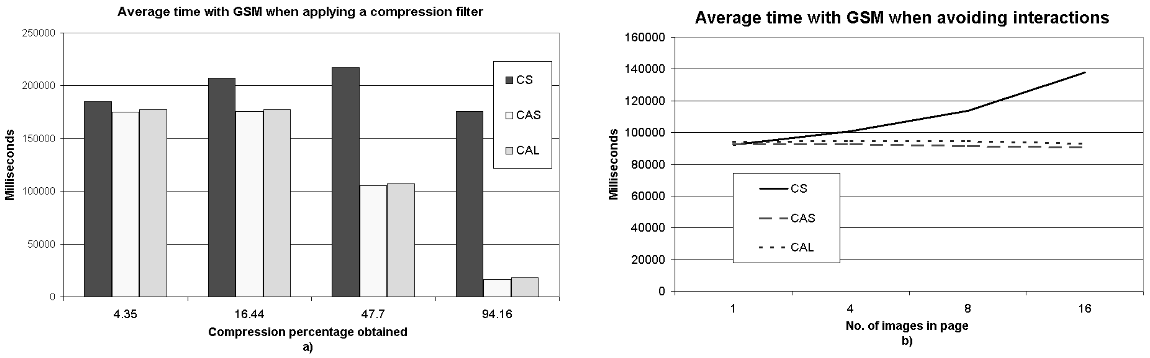

Figure 2. Comparison of the minimum time the MU has to be connected with each of the approaches.

proaches with the intermediary agent, the user only has to specify the URL address of the page. Then, the *majordomo* obtains the HTML file of the page along with the files of the images. Once the agent obtains all files, it puts them in a package and sends them together to the MU. Table 2 shows the characteristics of the pages used. Figure 2.b shows the significant improvement on the time needed to download those web pages when using the wireless media.

Table 2. Set of web pages used for test 2.

Page	No. of images	size of each image	No. of files to download	Aprox. data to download
Page 1	1	80 Kb	2	80 Kb
Page 2	4	20 Kb	5	80 Kb
Page 3	8	10 Kb	9	80 Kb
Page 4	16	5 Kb	17	80 Kb

4. Conclusions

In this paper, we present a new data service for the mobile user: the *data locker service*. This service allows its users to use extra storage space located in the fixed network, outside of their mobile computer but close to them. Besides providing an extension of the storage space of the mobile computer, the service gives to its users more autonomy. Autonomy means that users can decide when to work disconnected or connected to the wireless network, since they can always count on the locker space for storing both the messages sent to them by other users and the results of their requests. Moreover, the use of lockers reduces the communication cost.

References

[1] Ibm aglets workbench - home page. http://www.trl.ibm.co.jp/aglets/.

[2] Milojicic et al. MASIF, the OMG mobile agent system interoperability facility. In *Proceedings of Mobile Agents'98*, September 1998.

[3] Y. Amir, B. Awerbuch, and R. S. Borgstrom. A cost-benefit framework for online management of a metacomputing system. *The International Journal for Decision Support Systems, Elsevier Science. To appear*.

[4] Fox et al. Experience with top gun wingman: a proxy-based graphical web browser for the 3com palmpilot. In *Procs. of Middleware'98*, September 1998.

[5] R. Gray, D. Rus, and D. Kotz. Agent TCL: Targeting the needs of mobile computers. *IEEE Internet Computing*, 1997.

[6] A. Joseph, J. Tauber, and M. Kaashoek. Mobile computing with the Rover toolkit. *IEEE Transactions on Computers: Special Issue on Mobile Computing*, 46(3), March 1997.

[7] J. Kistler and M. Satyanarayanan. Disconnected operation in the Coda file system. *ACM Transactions on Computer Systems*, 10:3–25, 1992.

[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] S. Papastavrou, G. Samaras, and E. Pitoura. Mobile agents for WWW distributed database access. In *Proceedings of the International Conference on Data Engineering*, 1999.

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

[11] C. Tschudin. Mobile agent security. In *Intelligent Information Agents - Agent based information discovery and management on the Internet. Springer*, pages 431–445. M. Klusch, 1999.

[12] 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.

[13] 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.