Transparent extention of existing applications for mobile computing

Rimbert Rudisch
Telecooperation Group
University of Linz
rru@tk.uni-linz.ac.at

Michael Beigl
Telecooperation Office
University of Karlsruhe
michael@teco.uni-karlsruhe.de

Abstract

Mobile computing devices together with an ever growing land-based and wireless communication network infrastructure are the existing technical prerequisites for continuous access to networked services. This relieves users from being bound to their desktop computers and lets them spread out into the world. But using the different mobile facilities often requires a lot of special concern and there is no straight-forward solution towards an integrated mobile environment. This paper examines an approach to extend existing systems by transparently introducing a mobility service system to meet the special needs of mobile users. Typical mobile communication problems such as frequent disconnection, low bandwidth or a changing network environment are hidden embedding the user into a supporting overall networked infrastructure but still providing his familiar workplace environment.

Keywords: mobile computing, distributed systems, mobile communication, disconnection, low bandwidth, mobile architecture, mobility services, mobile infrastructure

1 Introduction

With upcoming *mobile computing* the term *information anywhere, anytime* has been coined, meaning the ability of a mobile user to adequately access information utilizing his portable computer independent of his physical position. Technical prerequisites enabling this *just-in-place information* have been developed during recent years: Powerful portable units and ever growing public communication networks including wireless communication facilities.

Accessing suitable information often requires online network access, especially when large or frequently updated data sources have to be queried. Sometimes the user has the possibility of attaching directly to a computer network, e.g. while at his desk or in a branch office. At home or in a hotel a user can get online connected via the public telephone network. When there is no land-bound possibility of connection, for example while traveling or out in the field,

the user can still utilize a wireless networking interface. But the new possibilities also cause a lot of concern like frequent line breaks of wireless connections or widely varying quality parameters and charges of the different kinds of connection lines.

In most cases the mobile devices will be integrated in some kind of networked infrastructure consisting of a number of well known fixed hosts with various resources and services available [IMI94]. Architectures like this facilitate the support of the mobile user and make the advantages of client/server-based applications available.

Our goal is to provide a transparent *mobility interface* for ubiquitous information access concerning about optimal utilization of the current available communication line, facilitating the changing of communication lines and hiding communication disruptions while preserving the familiar environment of the user. Besides this care should be taken, using a networked infrastructure consisting of fixed hosts, about localization, authentication and security issues.

The organization of this paper is as follows: *Section 2* considers general issues of an infrastructure for mobile computing as mentioned above and presents an architecture of a mobility service system. *Section 3* discusses aspects of how to integrate mobility services into existing computer systems. *Section 4* describes an implementation of a mobility service supporting ubiquitous data access through the WWW service. Finally, *section 5* summarizes the paper and gives an outlook to future work.

2 Construction of Mobile Computing Systems

We consider *mobile computing* as the field of managing the mobility of users, devices and data. Special emphasis has to be laid upon the issue of integrating the mobile entities in an overall infrastructure consisting of a fixed host network supplying adequate information access utilizing wired ports or wireless connections.

2.1 From distributed computing to mobile computing

Following [MUE92] distributed systems can be divided into three basic categories: *Distributed operating systems* like PANDA [ASS93], Clouds [BHJ88] and Amoeba [GRT90] are trying to optimize distributed programs and distributed services by using a special distributed system kernel. These low level based systems are very efficient in executing distributed programs and services, but are also very specialized according to these distributed problems. To enhance these systems there are often program development systems added, e.g the CC++ for Clouds. *Distributed programming systems* were first based on Remote Procedure Calls, like Sun-RPC [RFC1057] [SUN85] or DCE [OSF92]. Newer systems like DC++ [MOC93] [BEI95], cooC [IMM93] or OMG's CORBA [OMG92-11-1] [OMG93-7-2] [OMG95-3-xx] are based on distributed objects. These systems

are aimed to support the building of distributed applications. They are typically not as fast as the distributed operating systems according to the execution speed. However they could be integrated in existing environments and operating systems. Also the possibility of writing programs with higher degrees of parallelism could speed up special applications. *Distributed service systems* provide an access to a special service. Examples are distributed database systems and the X-Window system. These systems provide access to one service using a proprietary protocol.

Mobile systems have different requirements and restrictions, not yet taken into account by conventional distributed systems. Assumptions often implied when dealing with distributed systems are:

- *Fast communication lines*
  Communication lines are fast, at least faster than 1 MBit/s. Many Systems even need lines at a bandwidth of 10 MBit/s.

- *High reliable communication and low error rate*
  Communication has to have a high quality of service. This means, that line breaks are seldom and the quality of the communication signals is good enough to ensure the bandwidth being almost full available for data communication.

- *No long times of disconnection*
  One of the most important assumption in common distributed systems is, that reply times to a call are below a certain minimum. This is especially true for kernel-based distributed systems.

- *Appearance of unknown communication error is not very probable or impossible*
  The appearance of unknown communication errors is very difficult to handle. In fact, in most distributed environments, based on Local Area Networks or Wide Area Networks, there is a very low probability for such an error case.

- *Application focused*
  Distributed systems take less care of special user needs but lay emphasis on supporting distributed applications and services.

Mobile systems need to overcome these assumptions, but should also take previous goals of distributed systems into account. Here is a short summary of the basic problems which distributed systems are destined to solve [MUE92] [MUL91] and which are also relevant for mobile systems. The first one is that emphasis lies on *decentralizing* data. This implies the replication of data in a somehow chaotic and not foreseeable manner. *Cooperation*, especially the cooperation of users is another problem. This has to be handled differently to "normal" distributed systems, concerning the mostly lower bandwidth e.g. when sharing multimedia data. A third problem is *integration* of separated parts of an application. Sometimes even with low bandwidth spreading of an application is

useful, mainly when the accessing client computer doesn't have the specialized hardware or other resources to fulfill a certain task. *Distributed access* to resources is a basic demand for mobile distributed systems. In a mobile scenario resources can be found everywhere in the world, depending on unforeseeable constraints. Accessing the needed resource and managing of this distribution are central problems in the area of mobile computing.

2.2 Requirements of mobile systems

Integrating a mobile system into current computer systems is quite difficult, because of the many assumptions present systems make relating to the non-mobile behavior of the environment. Care has to be taken of where and how to integrate mobility services. Therefore first the requirements for such a mobile computing systems should be named.

- Transparent access to distributed data should allow remote data access without concerning the user. Informations and associated services are located with the help of services.
- Bandwidth depending services with focus on problems caused by low bandwidth should be provided. This means that applications react intelligent and provide the user a view of the data according to the current bandwidth. E.g. a picture viewer service can compress the picture with loss in case of low bandwidth.
- The system should be stable, i.e. no error "from the outside" can interfere the execution of an application. In special, communication errors have to be handled transparently by the mobile system without the interference of the user.
- Special emphasis has to be laid upon the issue of extendibility. There is a need for new services in the field of mobile computing. These mobility services will offer diverse features a mobile system should support and therefore should be addable.
- A common service structure for all mobility services should be defined. There should be a common set of functionality and a common communication protocol for all mobility services to share. Despite of the specialties every service inhabits, all services share some communication facilities and behavior to be able to react on communication events.
- Transparent integration in existing systems and applications should be an important goal. The mobility services should shadow everything concerned with mobile and distributed computing support from the applications.

According to these requirements a mobility service system is specified, which is presented in detail below.

2.3 A Mobility Service System Architecture

To overcome the shortcomings of previous distributed systems, a "mobile distributed architecture" is specified, considering the topics discussed above. Looking at the services we distinguish two classes of them: common mobility services providing basic functionality common to all services and special mobility services, implementing special tasks.

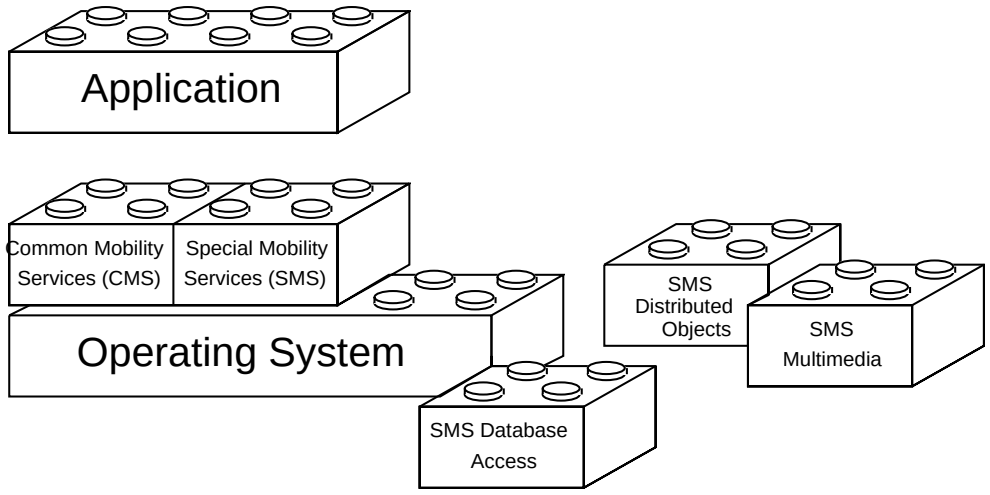

Figure 1: A Mobility Service System Architecture

For this reason the mobility service system architecture consists of two distinct blocks of distributed services (Figure 1). The block of *Common Mobility Services (CMS)*, supplies the basic mobility infrastructure. This block includes, among others, location service for special mobility services, connection handling service, authentication service and encryption service. This functions are basic for accessing network connection, routing, executing security tasks and watching for errors, especially line breaks. The CMS creates a channel from the CMS-client to a CMS-server running on a remote system providing resources for a special mobility service. At least one *Special Mobility Service (SMS)* providing some functionality to applications is loaded at the system. The SMS supports the access of remote services from the user's application and tunnels this requests to a SMS-server through the channels opened by the CMS. Please note that in the schematic representation of the mobile system (Figure 1) the mobility services provide the same interface to applications as the operating system does.

2.4 Scenario

This real-world scenario is presented in terms of clearifing the issues discussed in this section and evaluating the given approach. A typical scenario of mobile computing is the one of a traveling salesman. A traveling salesman enters his home office in the morning and gathers topical data and correspondence on his personal desktop computer. A visiting and working-plan for the day is created

based on this information. Relevant customer data is collected and copied onto his laptop computer. This could be automated using an intelligent caching mechanism, which refers to user and application profiles.

Afterwards the salesman visits the customer where a consultation takes place. Now presentation data needed but not cached on the computer can be accessed using a wireless phone. If possible, large pictures and similar data are compressed or even not transmitted at all to shorten access time and save money. Sometimes the salesman gets the possibility to connect at the customer's site to query some information from the local database to enhance his consultation. This introduces the need for supporting heterogeneous environments. A compiled offer or order is electronically forwarded immediately to the salesman's company.

Back at home in the evening he can connect to a land-based network, for example ISDN, to transfer customer data needed for the next consultation from his office onto his laptop or Home-PC.

As we have mentioned above several tasks have to be fulfilled by a mobility service system. The central task is the transparent access to data (the company's database) hiding the current communication line (Ethernet, wireless phone, ISDN). The access should be efficient and adapted to the current bandwidth. The system should be able to recover from a communication error without interference of the user. Another task is intelligent caching of data onto a laptop computer. Sophisticated profiles of the user and applications have to be created and are used to decide what to cache. The goal is that when a working plan for a day is presented to the system by a certain user, a set of data accesses is predicted and the according data is copied into a cache on the laptop computer. In a foreign environment services like user profiling, authentication and routing to the central information database are important to enable transparent data access.

3 "Mobilizing" existing systems

In this section ways of how to extend existing systems to introduce mobility support are discussed. But first, the already heavily used term "service" shall be clarified. As a *service* we define a set of specific functions usually executed as a distinct process. We term services implemented for needs of mobile computing as *mobility services*. As a special case we call *mobilized services* existing services which are subsequently extended to match the needs of mobile computing.

3.1 Approaches for the integration of mobile systems

Considering distributed systems, three ways to integrate support for mobile computing into a system can be identified. The first one is the *integration of mobility services into the kernel*. However this approach raises some problems, which are mainly:

- Network access is inefficient because of slow communication lines. E.g. moving a memory page from one computer to another is too slow in a typical GSM based wireless system with bandwidth of 9.6 kBit/s.
- Line breaks often cause a system to crash because it depends on getting important kernel information
- Transaction control could only be done in kernel terms, i.e. memory pages. This is too coarse grained when high error rates often require retransmission.
- Mobile computing needs some kind of "intelligent" support. This means the communication system must for example find another path to a special computer or even must find information on another location on his own. In case of low bandwidth the system may have to decide upon compression of data. For that reason the system needs to know the types of the data. This is impossible when handling unstructured kernel data.

A second possibility are *mobility services implemented by programming systems*. Introducing a mobility programming system would not allow to hide the changes from existing software. One of the problems is, that it is not possible to take advantage of mobility software when using it together with current available software. E.g. it is impossible for the mobile system to close connection between two requests in terms of saving line costs when one of the software parts in the system could not handle this transparently. This lacks the goal of transparently integrating the mobility system into current software systems.

As a third approach one could build a mobile system as *one special service*. But a mobile system as a special service is not sufficient enough for our system concept. As described in section two an overall supporting service architecture should be available. All special services have to satisfy these architecture and the according gain to fulfill the goal of transparency.

3.2 Mobilizing APIs

The goal is to integrate mobility support into existing systems without having to change available applications. The proposed way takes advantage of the mechanism of dynamic loading in micro-kernel systems, where just a very concise part, i.e. the kernel, stays resident in memory, while other parts of the operating systems, like drivers, are loaded during runtime and only when they are required.

Being parts of the operating system those unsteady software modules have a standardized programming interface (API) so applications are able to employ the provided functionality. Being dynamically loaded, this software modules can be exchanged without the need of changing the whole system. New modules receive the same API-Calls from applications as the old ones. This way the functionality of an operating system can be enhanced transparently to the applications. Introducing the term " mobilizing APIs" we want to stress the

fact, that mobility is supported by parts of the operating system without any change to their programming interface for applications.

3.3 Integration of the Mobility Service System Architecture

The mobility service system architecture presented in section 2.3 can be viewed as a kind of distributed system based upon the operation system level. Special "high level" drivers can be replaced to enhance the system for mobile needs. Basically any system service can be mobilized, i.e. providing the ability of transparent access from nearly everywhere.

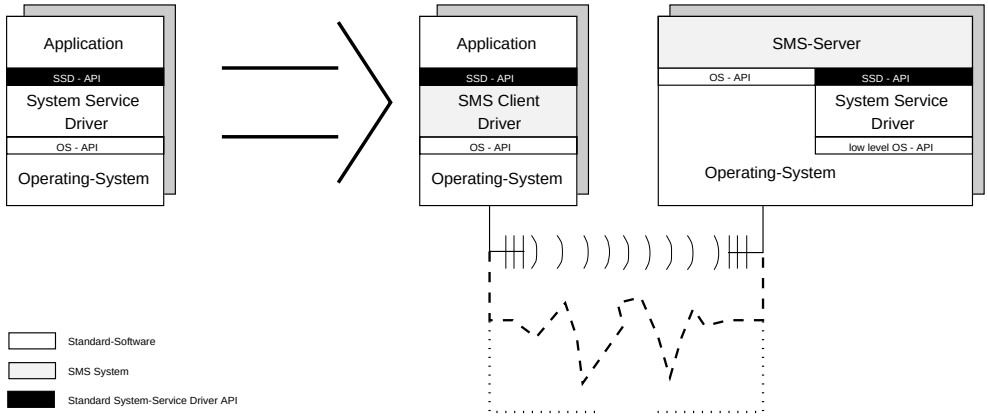

Figure 2: Mobile driver architecture

Figure 2 presents a schematic overview of the described solution. The block on the very right side shows the server part. Here the service requests from the client are received and handled. This part consists of the *SMS server*, which tunnels the client's requests received over the operating systems network interface to the system service driver. The block on the in the middle shows the client part, where the requests from applications for system services reach the *SMS Client Driver*, which forwards them over the net. Client and server can be connected by different and varying communication lines. On the client side the applications access the mobile client using the normal system service driver API.

4 Mobile WWW

4.1 Architecture

As mentioned above our mobility service system consists of two different kind of services: Services providing functionality to meet the basic requirements (Common Mobility Services, CMS), and services implemented to handle more specialized, high level tasks (Special Mobility Services, SMS). The proposed system architecture for the support of mobility is based on the mobility services as its building blocks. Picture 3 is an example of an exiting implementation of such a system.

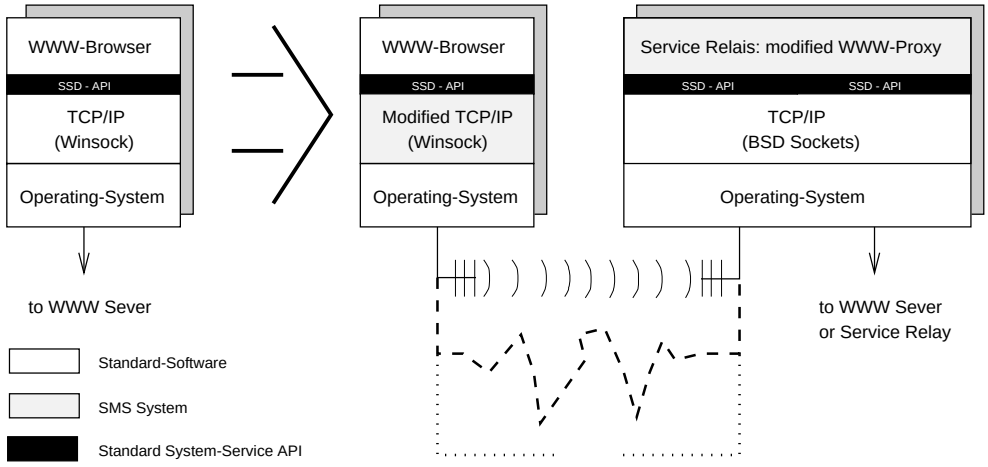

Figure 3: Mobile Information Service with WWW

The presented architecture is not restricted to the task of mobilizing WWW services. On the basis for this architecture we have previously build a Mobile-ODBC for accessing SQL-Server from mobile stations. Mobilized services take advantage of the functionality provided by the CMS. Needed by our example system is a quality-of-service measuring tool, that takes care about the existing quality-of-service and the user-profile-service, which propagates the users preferences. Both services allow the system to react "intelligent" to the users need and to environment restrictions; therefore the system can provide a satisfying service both in time and money. Beside of that we use our CMS for connection-oriented lines.
To give an closer look at our mobility service system one SMS and CMS Service that is actually implemented should be explained in more detail below.

4.2 Common Mobility Service: The Connection Management Service

Using mobile communications one encounters well known problems such as disconnection, varying quality of communication facilities and varying line costs [FOR94] are to be solved.
Concerning disconnection, we have to deal with two different situations: Intentional disconnection when changing the communication line and spurious disruption during a transmission. Both of this cases are dealt with by the implemented service: Before starting a transmission, the different communication ports are sequentially checked for availability in the order of their quality. In this way the optimal available communication line is chosen. For example first we try to get connected via the ethernet adapter and in case of failure we try to connect using the modem.
During ongoing transmission, the connection line is constantly watched and if a disruption is noticed an automatic reconnection procedure is initiated. If the same connection cannot be established again, a rerouting process is started, for example the system is trying to connect to another well-known dial-up point.

4.3 Special Mobility Service: Compression for WWW

Another problem with mobile communication is the low transfer rate of communication lines and the high price of communication costs. Therefore we use online compression to reduce the amount of data. Especially lossy compression of graphic, video and audio lead us to the needed compression rates of 1:10 - 1:100.
Graphics, video and sound files that are included in the document are converted according to the values of the requested quality of service. For example it is possible that the client tells the server only to send dithered pictures with a given maximum size; the user profile of service parameters describe for every class (graphic, video, sound) a compression rate the client wants to receive.

5 Conclusion

In this paper first distributed and mobile systems were compared to benefit from approved paradigms but also learn from their shortcomings. As a result a mobility service system architecture was defined, integrating mobile entities into supporting infrastructures. Next the integration of mobility support into existing systems by "mobilizing" parts of their API, thus shadowing the changes to existing applications was examined. Finally "Mobile WWW" was presented as an implementation of the introduced architecture, which allows mobile data access without having to change the applications.

In the future the research is extended to refine the already existent mobility support and to mobilize other services in the context of the MS-Windows NT operating system such as ODBC.

References

[ASS93] H.Assenmacher, T. Breitback, P. Buhler, V. Hübsch, R. Schwarz
The PANDA System Architecure - A Pico-Kernel Approach
Proc. of the Fourth Workshop on Future Trends of Distributed Computing Systems, 1993

[BEI95] Michael Beigl
Entwicklung einer CORBA-konformen Sprache mit Übersetzer für DC++
Diplomarbeit an der Fakultät für Informatik der Universität Karlsruhe, 1995 (in German)

[BHJ88] J. M. Bernabéau-Aubán, P. W. Hutto, M. Yousef, M. Y. A. Khalidi, M. Ahmad, W. F. Appelbe, P. Dasgupta, R. J. LeBlanc, U. Ramachandran
Clouds - A Distributed, Object-Based Operating System, Architecture and Kernel Implementation
UNIX Systems User Group Conf., 1988

[FOR94] Gorge H. Forman, John Zahorjan
The Challenges of Mobile Computing
Tech Report 93-11-03
Computer Science and Engineering, University of Wahington
Accepted for publication in IEEE Computer

[GRT90] S. J. Mullender, G. van Rossum, A. S. Tanenbaum, R. van Renesse, H. van Staveren
Amoeba: A Distributed Operating System for the 1990s
IEEE Computer, vol.23, 1990

[IMI94] Tomasz Imielinski and B. R. Badrinath
Mobile Wireless Computing
Communications of the ACM 37, 1994

[IMM93] A. Inoue, K. Maeda, A. Morishita, N. Sawashima, I. Tomoda, R. Trehan
Concurrent Object Oriented C (cooC)
ACM Sigplan Notices, 1993, February

[MUL91] Sape Mullender, ed.
Distributed Systems
ACM Press, 1991

[MUE92] M. Mülhäuser, A. Schill
Software Engineering für verteilte Anwendugen
Springer-Verlag, 1992

[MOC93] Markus U. Mock, Alexander B. Schill
DC++: Distributed Object-Oriented System Support on Top of OSF DCE
Distributed Systems Engineering Journal 1993

[OMG95-3-xx] CORBA 2.0/Interoperatbility
Open Management Group, Framingham, USA, 1995

[OMG93-7-2] ORB Architecture document
Andrew Watson
Open Management Group, Framingham, USA, 1993

[OMG92-11-1] Object Management Architecture Guide, Revision 2.0
Open Management Group, Framingham, USA, 1992

[OSF92] Introduction to DCE
Open Software Foundation, Cambridge, USA, 1992

[RFC1057] Sun Microsystems Inc.
RPC: Remote Procedure Call Protocol Specification Version 2
IETF, 1988

[SUN85] Remote Procedure Call Specification
SUN Microsystems, Inc., 1985