Mobile Host: A feasibility analysis of mobile Web Service provisioning

Satish Srirama[1], Matthias Jarke[1,2], and Wolfgang Prinz[1,2]

[1] RWTH Aachen, Informatik V
Ahornstr.55, 52056 Aachen, Germany
[2] Fraunhofer FIT
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{srirama, jarke}@i5.informatik.rwth-aachen.de
wolfgang.prinz@fit.fraunhofer.de

Abstract. This paper, discusses the feasibility of mobile Web Service provisioning from Smart Phones, confluencing two major recent trends in distributed information systems engineering: the evolution from static content via personalized adaptive information provisioning to Web Services, and the emergence of mobile terminals with sufficient speed to serve as parts of information systems. The research of mobile Web Services has been mainly concentrated on Web Service client systems. Mobile Web Service provisioning was ignored, since a realization seemed beyond the resource capabilities of present mobile terminals. Complementing this work, here we discuss one such prototype of a mobile Web Service provider, developed and evaluated in cooperation with a major mobile phone vendor. We also present a detailed performance analysis of this Mobile Host, proving its feasibility.

1 Introduction

Web Services are software components that can be accessed over the Internet using established web mechanisms, XML-based open standards and transport protocols such as SOAP [1] and HTTP [2]. Public interfaces of Web Services are described using Web Service Description Language (WSDL) [3]. Examples of Web Services range from simple requests, such as stock quotes or user authentication, to more complex tasks, such as comparing and purchasing items over the Internet.

With the introduction of Third and Interim Generation mobile communication technologies in the cellular domain like UMTS, GPRS/EDGE [4], the speed of wireless data transmission has increased significantly. Also processing power and device capabilities of mobile phones have increased drastically, thereby enabling better applications and usage of mobile devices in different application domains.

Combining these developments it is a logical next step to turn mobile devices into wireless Web Service requestors (clients) and even providers (Mobile Hosts). This enables communication via open XML Web Service interfaces and standardized protocols also on the radio link, where today still proprietary and application- and terminal-specific interfaces are required. Mobile Web Services lead to manifold opportunities to mobile operators, wireless equipment vendors, third-party application developers, and end users. It is easy to imagine that in the future mobile applications based on Web Services will generate a large percentage of all Web Service requests, and the first such solutions are currently appearing on the market. [5–7]

Performance of mobile Web Service clients was extensively studied, proving their feasibility also with commercial applications [8]; even though some questions were raised having such a verbose SOAP communication over the slow wireless networks [9]. Our study mainly focuses at extending this study also to mobile Web Service providers, proving their feasibility. In this paper, we explore this idea of Mobile Hosts and their performance, based on our experiences from a cooperation project with a major mobile & telecom service provider in which the prototype of such a Mobile Host was developed and tested on a smart mobile phone. The paper is organized as follows:

Section 2, introduces the concept and applications of mobile Web Service provisioning. Section 3 briefly describes the theoretical performance model for mobile Web Service invocation. Section 4 discusses the means of accessing the Mobile Host in mobile operator network from Internet and section 5 describes the detailed performance analysis conducted and its evaluation results. Finally section 6 discusses conclusions and further research directions.

2 Mobile Web Service Provisioning

The basic architecture for Web Services is built upon its three components: Service Requestor (Client), Service Provider and Service Registry. The service provider publishes its Web Services with the service registry. The service requestor later searches ("Find") the UDDI registry [10] for the services, and the UDDI compatible service registry refers the respective WSDL for the service. The service requestor accesses the described Web Service, using SOAP. Similar to this, the basic architecture of the mobile terminal as Web Service provider can be established with the Web Service provider ("Mobile Host") being implemented on the smart phone.

The Mobile Host has been developed as a Web Service handler built on top of a normal Web server. The Web Service requests sent by HTTP tunneling are diverted and handled by the Web Service handler. The Mobile Host was developed in PersonalJava [11] on a SonyEricsson P800 Smart Phone. The footprint of our fully functional prototype is only 130 KB. Open source kSOAP [12] was used for creating and handling the SOAP messages. Most recently kSOAP is redesigned to kSOAP 2 [13] and the Mobile Host was upgraded to this new version, without much effort. But most of the performance analysis results discussed in this paper are with respect to kSOAP. [6]

Even though the Web Service provider is implemented on the Smart Phone, the standard WSDL can be used to describe the services, and the standard

UDDI registry can be used for publishing and un-publishing the services. This of course presents the challenge to design the mobile terminal with the same general architecture as on any standard desktop system, even under the low-resource considerations of the Smart Phone. An alternative for the mobile Web Service discovery is being studied, where we are trying to realize Mobile Host in a Peer to Peer (P2P) network, there by leveraging the advertising and searching of WSDL to the P2P network. A detailed explanation of this procedure is beyond the scope of this paper.

Mobile Host opens up a new set of applications that has not been explored very much up to now, since a realization seemed beyond the resource capabilities of present mobile terminals. It can be used in domains like location based services, mobile community support and pervasive gaming etc. From a commercial viewpoint, it also renders possibility for small mobile operators to set up their own mobile Web Service business without resorting to stationary office structures. [14]

Of course, this additional flexibility generates a large number of interesting research questions which need further exploration, the immediate topics of interest being checking the feasibility of Mobile Host in handling reasonable services. The following sections will explain the considered performance analysis model and the results.

3 Performance Model

The developed Mobile Host was extensively tested for performance issues like the memory load, server-processing load etc. For this evaluation, a simple request-response scenario was started, in which a standalone Axis [15] client (The client can be any standard Web Service client) program was developed, which accesses the Mobile Host as a Web Service requester. The client calls for different services deployed on the Mobile Host and the performance of the Mobile Host is observed while it is processing the Web Service request. The considered services and the access methods are discussed in the latter sections. Figure 1 shows different operations performed and time components that constitute one complete Web Service invocation cycle.

The client initiates the call for the Web Service and the Mobile Host processes the request, populates the response, and sends response back to the client. The total time taken for this mobile Web Service invocation (T_{mwsp}) constitutes, the time taken by client for constructing valid SOAP message (T_{cc}), the time taken to transmit the SOAP request to Mobile Host (T_{reqt}), the time taken for de-serializing the XML based SOAP message to SOAPEnvelope object (T_{sd}), the time taken by the Mobile Host to execute the respective business logic and to populate the response ($T_{process}$), the time taken for serializing the SOAPEnvelope object back to XML data streams (T_{ss}), the time taken to transmit the SOAP response back to the client (T_{rest}) and lastly the time taken by the client to process the response (T_{cp}). The invocation process is shown in figure 1 and the total time taken for the mobile Web Service invocation is given in the following

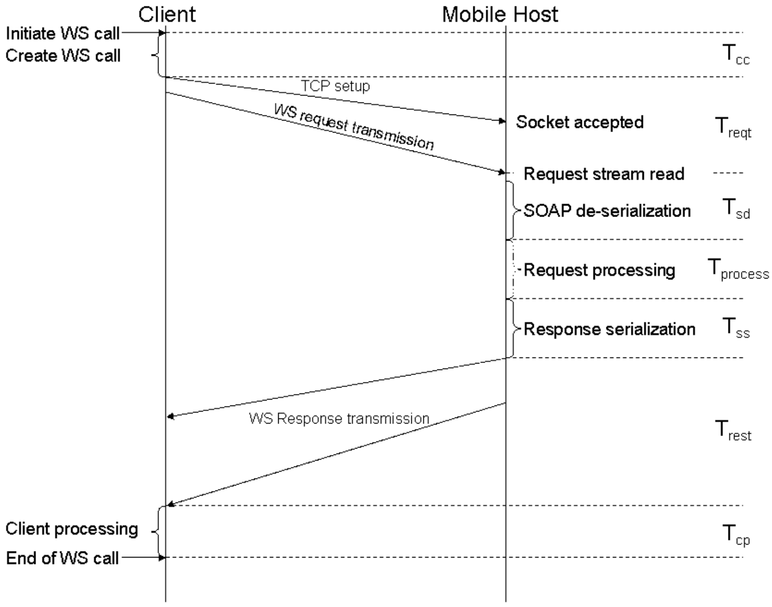

Fig. 1. Mobile Web Service invocation: operations and time stamps

equation.

$$T_{mwsp} = T_{cc} + T_{reqt} + T_{sd} + T_{process} + T_{ss} + T_{rest} + T_{cp} \quad (1)$$

The request and response messages are transferred to the Mobile Host in the form of TCP packets. So some delay could be caused by packet loss, TCP congestion control etc. The delay is shown in the figure as the slanting lines for request and response transmissions. T_{tcp} represents this delay caused by the transmission protocol.

$$T_{tcp} = \delta_{reqt} + \delta_{rest} \quad (2)$$

where δ_{reqt}, δ_{rest} are the respective propagation delays caused while transmitting the SOAP request and response messages. In the proposed performance model the transmission times (T_{reqt}, T_{rest}) also include these tcp delays and an estimation of the tcp delays is not specifically observed.

4 Mobile Terminal Access

Once a Web Service is developed and deployed with the Mobile Host, the mobile terminal, that is registered and connected within the mobile operator network, requires some means of identification and addressing, that allows the Web Service to be accessible also from Internet.

Generally, computers and devices in a TCP/IP network are identified using an IP address. The IP address, that is required for the data transfer to and from Smart Phones (as for any other IP communication client as Web servers, Intranet workstations, etc.), is assigned during the communication configuration phase. Typically, the IP address assigned to mobile devices using GPRS is only temporarily available, and is known only within the mobile operators network, which makes it difficult to use the IP address in the client applications.

The study has identified different means of resolving the IP address in HSCSD (High-Speed Circuit Switched Data) dial-up connection, GPRS (General Packet Radio Service) environments and thereby making the data transmission with a mobile terminal, possible. Here we discuss two of these identified methods.

4.1 HSCSD

Figure 2 illustrates the architecture used to connect the Mobile Host to the prototyping network using a HSCSD dial-up connection. In this architecture a HSCSD connection is established between the mobile terminal and the prototyping network, which is connected to the Internet. The connection uses a Public Land Mobile Network (PLMN) and the Public Switch Telephone Network (PSTN / ISDN) for making the data call to the server. The connection is setup by using PPP (Point-to-Point Protocol) over a circuit-switched data call to a modem that is connected to one of the servers in the network. On top of this PPP link a TCP/IP end-to-end connection between the mobile terminal and the dial-in server is established. Hence, as long as the data call persists, the mobile terminal can be addressed using the IP address assigned to it by the dial-in server. Thus the Web Service deployed on the mobile terminal can be accessed from any client within the network environment.

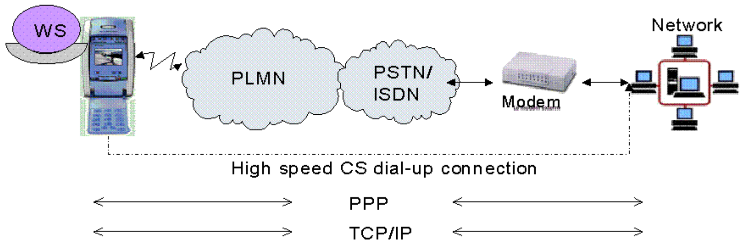

Fig. 2. Architecture for an end-to-end TCP/IP connection between the mobile terminal and the prototyping network using HSCSD connection

Using an appropriate NAT configuration, the mobile Web Service can be accessed by any service requestor from the Internet. Using the NAT, the network provides a DNS name for the Mobile Host. The only requirement towards the

PPP daemon is that the mobile terminal should always receive the same IP address when it connects to the dial-in server.

The main drawback with the HSCSD solution is the circuit switched connection, which would have to persist as long as the Mobile Host should be available for the access of its Web Services. The billing of circuit switched data connections is based on the time the connection persists, not on the amount of data transmitted across the network. This makes this scenario unfeasible for commercial purposes. Volume based charging is a major advantage enabled by GPRS.

4.2 GPRS

Once the GPRS connection is established the mobile can be identified by the temporary IP provided by the mobile operator network. It is also feasible to have a public IP for the mobile terminal, a feature provided by very few operators today. The operational setup for accessing the mobile terminal in a GPRS network is given in figure 3. The mobile TCP/IP connection between the Web Service client and the Mobile Host is deployed on top of a GPRS link into the mobile operator network. From there the traffic is routed through the Internet to/from the Web Service client.

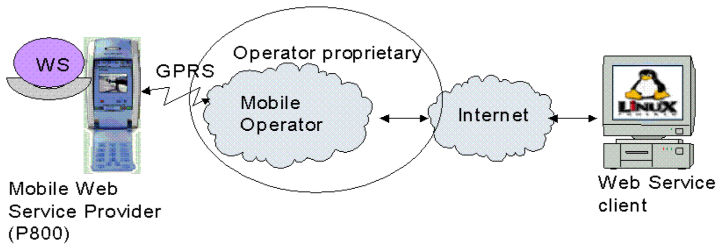

Fig. 3. The operational setup of Mobile Host in a live GPRS environment

The problem of addressing each mobile node with IP is not a big issue and it could be solved with Mobile IP version 6(Mobile IPv6). [16]

5 Performance Evaluation

To prove the feasibility of the Mobile Host, different services were developed and deployed with the Mobile Host on P800 Smart Phone. The following subsections first describe these services and then explain the test setup, experiments conducted and obtained results.

5.1 Test Case Services

Mobile Photo Album Service Todays high-end mobile terminals become more and more advanced, and are generally being equipped with an integrated digital camera. The photographs taken with these mobile phones can later be uploaded or transferred to PCs through cables or by using wireless methods like Infrared or Bluetooth. Using currently available technologies, if a user wants to publish the photographs he had taken with the mobile terminal to the public or friends, he has to upload the photos to a Web server, from which they can be accessed. The user can also send the images through MMS or some other means of messaging to the clients. Here the mobile owner bears the payment for the communication between his Smart Phone and the Web server or the receivers device. With a mobile Web Service provider, implemented and deployed on the Smart Phone, interested people can access the Mobile Host using a standard Web Service client or a Web client, and can browse through the pictures they are interested in. Here the responsibility for payment shifts to the actual clients, browsing the pictures provided by the Mobile Host. The service is comparable to any other online image album service, but implemented on the mobile terminal.

Also for the performance evaluation, the service was quite useful. The response of the service is comparitively large (approximately 40KB) and this gives a large scope for observing the effects of different parameters like transmission delays, the encoding performed on the response messages, the actual service delay, and etc. on the performance of Mobile Host. For the test cases, 15 different images were selected with memory sizes ranging from 3KB to 100KB.

Location (GPS) Data Provisioning Service This dedicated Web Service provides the exact location information of the mobile terminal, such as GPS (Global Positioning System) data [17]. The service uses a Socket GPS receiver for getting the GPS co-ordinates. The external device is connected to the Smart Phone via Bluetooth. The service returns just a small string (approximately 2 KB) containing the GPS data as the response, which gives the scope for observing the behavior of the Mobile Host under concurrent requests from multiple clients, there by observing the robustness of the Mobile Host.

Many application were developed and demonstrated using these two services, for example in a distress call, the mobile terminal could provide a geographical description of its location along with location details. Another interesting application scenario involves the co-ordination between journalists and their respective organizations. The scenario is illustrated in figure 4. Journalists can be at different locations across the globe, covering different events like the sport events, conferences, etc. An editor can always keep track of the location of "his" journalists and the content they have gathered. He can browse through the pictures taken by the journalist at any instance. Standard client applications can be developed for the editor, which synchronize the information stored by editor and data at the Mobile Host. The key difference to the more traditional solutions where journalists upload their contents to a server held by the Editor is that parallel access to the Mobile Host by both the journalist and the editor is possible; even other journalists in the team can look at the mobile information thus better synchronizing their activities, e.g. in the coverage of some major distributed event. Thus, the journalists can concentrate more on their job of collecting, as they dont have to upload the data, every time they get something interesting. The data can later be synchronized with a server for archives, when the journalists are free and off the event site.

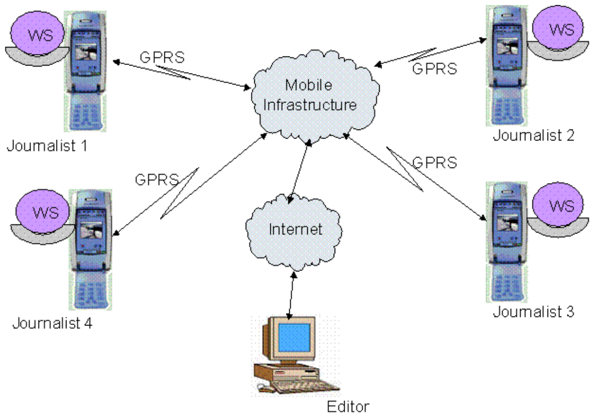

Fig. 4. The Mobile Host in collaborative journalism scenario

Applications using location data provisioning service are generally useful in the people-to-people-to-geographical-places(P3 Systems) in the social networks domain [18]. These location-aware community sytems throw many privacy concerns like possibility of "stalking" or simple violations of users' desire for privacy. Our current research in this domain also includes these security concerns and they can be eliminated with user intervened authorization, where the mobile user, providing the service can validate the authenticity of the location data requester. Also the process can be automated with a rule based intermediary component called "Mobile Web Services Mediation Framework", which maintains the individual user profiles and personalization settings. A detailed discussion of these security issues is beyond the scope of this paper. Apart from this, some research results show that people are less concerned about their location being tracked, as long as they find the service useful [19].

Apart from the Mobile photo album service and Location data provisioning service, some more basic services like echo, ls services and etc. were also used in the performance analysis.

5.2 Test setup and results

For the evaluation, a standalone Axis Web Service client was used to access the services deployed on the Mobile Host. Both the HSCSD and GPRS mobile terminal access methods discussed in section 4 were used for identifying the Mobile Host. First the mobile picture service was used for calculating the SOAP processing delay of the server. The results showed a significant difference (approximately 20%) between the time taken for Web Service access and the normal HTTP access. Figure 5 shows these times for the HSCSD connection.

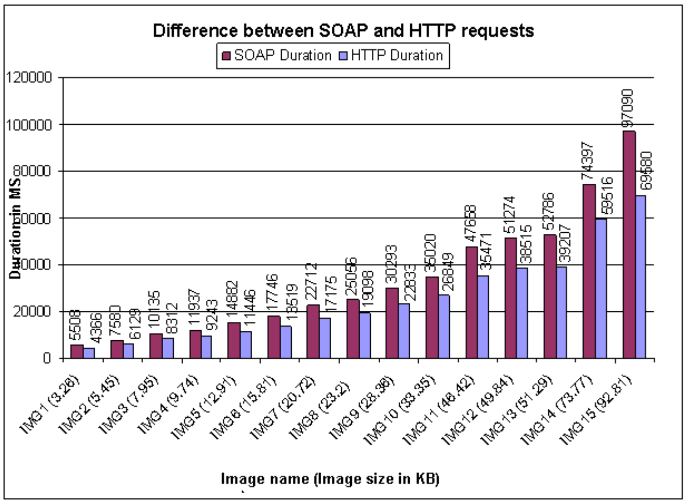

Fig. 5. Difference between round-trip durations for the Web Service and HTTP requests

The SOAP overhead and the Base64 encoding performed on the images before serialization of the response, has caused the size of the response to increase by more than 50%. The actual SOAP overhead caused to the size of the response is observed to be only 578 bytes. This was observed using the Echo service with a single character request. The increase in the size has increased the transmission delay and there by increasing the delay in response.

In order to identify the actual times taken for different activities on the Mobile Host like T_{sd}, T_{ss}, T_{rest} etc., the location based service was requested by the client and the time stamps were taken at the Mobile Host while it was processing the request. These time stamps were later processed to get the operational time delays. Figure 6 shows the time delays of different activities, for the Location data provisioning service using the GPRS connection.

The observation of these results suggest that the total WS processing time as a combination of de-serialization, SOAP processing and serialization, at the

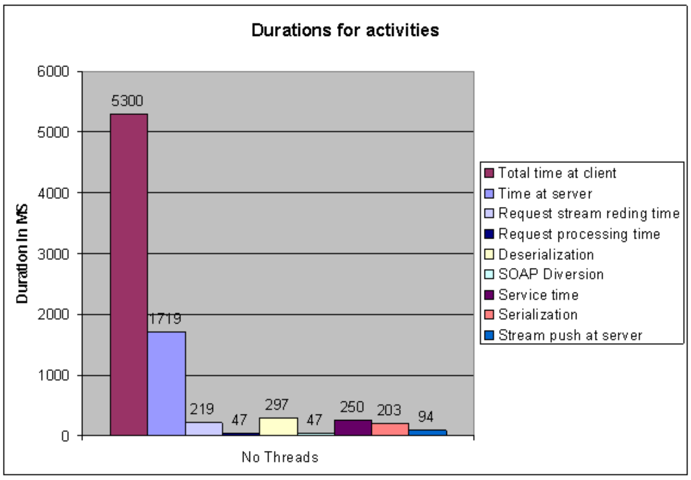

Fig. 6. Time stamps for the GPS data provisioning service

Mobile Host, is only a small fraction of the total request-response time (<10%) and rest all being mostly transmission delay. The test also conducted with the mobile picture service revealed that WS processing time is still negligible and the total response time linearly increased with the size of the image.

In terms of performance of the Mobile Host, the key question was whether a reasonable number of clients could be supported with an overhead that would not prevent the main mobile user from using his or her Smart Phone in the normal fashion (either to supply the services or just for usual local phone functions). Concurrent requests were generated for the services deployed on the Mobile Host, simulating multiple clients. The results of this regression analysis for checking the scalability of the Mobile Host are very encouraging and the Mobile Host was successful in handling 8 concurrent accesses for reasonable service like location data provisioning service. But it was observed that the concurrent access can affect the Mobile Hosts ability to access internal and external resources. The comparison of the time stamps for the GPS data provisioning in unique and concurrent access revealed that the only drastic difference was at the $T_{process}$.

The study of the memory footprints revealed that memory usage was not a problem with Mobile Host, as most of the time, the amount of free memory was at least 20% of the total memory allocated for the JVM (max value approximately 330 KB) and the "Out of Memory error" was never encountered during the execution of the tests.

Apart from these results, the comparison of results for HSCSD and GPRS connections suggested that the increase in transmission rates can increase the processing capability of the Mobile Host. Approximately 200 data traces were observed as the experiments were repeated several times in order to have statistically valid results.

6 Conclusion and future work

In this paper, we analyzed the feasibility of mobile Web Service provider for the Smart Phones. First we introduced the concept and the developed Mobile Host. Once the Mobile Host prototype was developed, it was extensively tested considering performance aspects in different real-time working environments/conditions. The evaluation clearly showed that service delivery as well as service administration can be done with reasonable ergonomic quality by normal mobile phone users. As the most important result, it turns out that the total WS processing time at the Mobile Host is only a small fraction of the total request-response time (<10%) and rest all being transmission delay.

Next steps on the technical side include a broader and more detailed, component-oriented performance analyses and means of improving the performance, an approach for discovering the services deployed on the Mobile Host; especially in a peer-to-peer network, and perhaps most importantly a detailed study of the QoS implications of this approach. Equally important is the study of specific application domains and usability analysis; our research mainly focuses on mobile community support and pervasive gaming.

Acknowledgment

The work is supported by German Research Foundation (DFG) as part of the Graduate School "Software for Mobile Communication Systems" at RWTH Aachen University. The authors would also like to thank M. Gerdes and R. Levenshteyn of Ericsson Research for their help and support.

References

1. W3C: SOAP, Simple Object Access Protocol, version 1.1, http://www.w3.org/TR/SOAP (2004)
2. HTTP: Hypertext Transfer Protocol version 1.1. IETF RFC 2616, http://www.ietf.org/rfc/rfc2616.txt (1999)
3. W3C: WSDL, Web Services Description Language, version 1.1, http://www.w3.org/TR/wsdl (2004)
4. GSM World: GSM - The Wireless Evolution, http://www.gsmworld.com/technology/index.shtml (2006)
5. JSR 172: J2ME Web Services Specification. (Java community process)
6. Srirama, S., Jarke, M., Prinz, W.: Mobile Web Service Provisioning. In: Int. Conf. on Internet and Web Applications and Services (ICIW06), IEEE Computer Society (2006)
7. Balani, N.: Deliver Web Services to mobile apps. IBM developerWorks (2003)
8. Tian, M., Voigt, T., Naumowicz, T., Ritter, H., Schiller, J.: Performance considerations for mobile web services. Elsevier Computer Communications Journal **27** (2004) 1097–1105
9. Laukkanen, M., Helin, H.: Web Services in wireless networks: What happened to the performance. In: Int. Conf. on Web Services ICWS 03, CSREA Press (2003) 278–284

10. UDDI: The Universal Description, Discovery and Integration, http://www.uddi.org/ (2004)
11. JCP: PersonalJava application environment specification, version 1.2a. SUN Developer Network, http://java.sun.com/products/personaljava/index.jsp (2000)
12. kSOAP: A open source SOAP implementation for kVM, http://ksoap.enhydra.org/ (2004)
13. kSOAP2, http://kobjects.org/ (2006)
14. Srirama, S.: Concept, implementation and performance analysis of mobile Web Service provisioning for Smart Phones (2004)
15. Axis 1.2: Apache implementation of the SOAP. Apache Software Foundation, http://ws.apache.org/axis/ (2004)
16. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. IETF (2002)
17. GPS: Global Positioning System: Data format . U.S. Coast Guard Navigation Center, http://www.navcen.uscg.gov/pubs/gps/sigspec/gpssps1.pdf (2004)
18. Jones, Q., Grandhi, S.A.: P3 Systems: Putting the place back into Social Networks. In: IEEE Internet Computing. Volume 09. (2005) 38–46
19. Barkhuus, L., Dey, A.: Location-Based Services for Mobile Telephony: a study of users privacy concerns. In: Proceedings of Interact 2003, ACM Press (2003) 709–712
20. W3C: Web Services Activity, http://www.w3.org/2002/ws/ (2004)