

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2005 Proceedings

Americas Conference on Information Systems
(AMCIS)

2005

An Agent Enabled System for Personalizing Wireless Mobile Services

Qusay H. Mahmoud

University of Guelph, qmahmoud@cis.uoguelph.ca

Zhixin Wang

University of Guelph, zhi@uoguelph.ca

Follow this and additional works at: <http://aisel.aisnet.org/amcis2005>

Recommended Citation

Mahmoud, Qusay H. and Wang, Zhixin, "An Agent Enabled System for Personalizing Wireless Mobile Services" (2005). *AMCIS 2005 Proceedings*. 227.

<http://aisel.aisnet.org/amcis2005/227>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

An Agent-Enabled System for Personalizing Wireless Mobile Services

Qusay H. Mahmoud

Dept. of Computing & Information Science
University of Guelph, Guelph, ON, Canada
qmahmoud@cis.uoguelph.ca

Zhixin Wang

Dept. of Computing & Information Science
University of Guelph, Guelph, ON, Canada
zhi@uoguelph.ca

ABSTRACT

Handheld wireless devices such as cellular phones and personal digital assistants (PDAs) have limited memory, storage, and processing power. In addition, small screens and limited input facilities make entering information tedious. It is therefore important that wireless mobile applications optimize resource usage and minimize input effort imposed on the user. One way is to download to the client only the information most relevant to the user, then present that information effectively, taking into account the user's preferences and history as well as the task at hand. This personalization approach minimizes the information to be displayed. In this paper, we present a mobile agent-based system for personalizing mobile services; we use mobile agents simply because such autonomous software entities have characteristics that can benefit mobile devices and the wireless environment. We introduce the tiered architecture of the proposed system and the functions of the different components; then we discuss how we use the Composite/Capabilities Preferences/Profile (CC/PP) in personalizing wireless mobile services. A proof of concept implementation has been developed using Java technologies.

Keywords

Mobile agents, CC/PP, context information, J2ME, WAP, wireless mobile services

INTRODUCTION

Mobile services are wireless mobile computing applications and services that can be either pushed to users' handhelds wireless devices (push mobile services) or downloaded and installed, over the air, on handheld devices (pull mobile services). Several technologies are available for developing mobile services; the different technologies and solutions can be classified into two categories: (1) *Browser-based*: applications are developed using a markup language. This is similar to the current desktop browser model where the device is equipped with a browser. The Wireless Application Protocol (WAP) [17] follows this approach where the Wireless Markup Language (WML) and XHTML are used for developing mobile services. And (2) *Native*: compiled applications where the device has a runtime environment to execute applications. Highly interactive mobile services, such as games, are only possible with this model. Such applications are developed using the Java 2 Micro Edition (J2ME) [9]. Another category is the hybrid application model that aims to incorporate the best aspects of both application models. In this model, the browser is used to allow the user to enter URLs to download native applications from remote servers, and the runtime environment is used to let these applications run on the device.

Handheld devices operate in a wireless environment that imposes constraints such as: wireless network environments are unreliable and bandwidth is low, and the very mobility of devices increases the risk that a connection will be lost or degraded. More importantly, mobile services must work within the daunting constraints of the devices themselves, which include: memory, processing power, input capabilities, and size of display. The display might be as small as 96 pixels wide by 54 pixels high and 1 bit deep (black and white), and therefore the amount of information that can be squeezed into such a tight screen is severely limited. Information overload [13] is a problem on much larger displays, where desktop users face too much information such as news and advertisements most of which is irrelevant to users. It is therefore important that wireless applications optimize resource usage and minimize input effort imposed on the user. One way to achieve this is to use *personalization*, which can be defined as the combined use of technology and customer information to tailor electronic commerce interactions between a business and each individual customer [14]. Using this technique and to minimize the information to be displayed on mobile devices, one can download to the client only the information most relevant to the user, then present that information effectively, taking into account the user's preferences and history as well as the task at hand. This is easier said than done! The task of designing a system for delivering personalized mobile services is complex because several issues need to be considered simultaneously. For example, what kind of content should the system select for mobile customers in order to satisfy their requirements? How to protect the user's personal information and privacy [16]? What kind

of format should we use to show the contents selected for users according to their different devices? And finally, given the multiple technologies used on mobile devices, what technology should be used to build personalized systems?

In order to provide high quality personalized mobile services to customers, the personalization system should consider customer's context situation. In other words, the system should use context to filter irrelevant information and/or services to customers. For example, the system should adapt the personalized services to customers' location, device characteristics, and adapt personalized content to different user interfaces. Our system is capable of delivering personalized mobile services to multiple clients as it supports both, push and pull technologies [15]. In other words, the system is designed to support WAP-enabled as well as J2ME-enabled devices. The system is agent-enabled in the sense that mobile software agents are responsible for discovering, filtering, and delivering personalized services.

Mobile Agents for Wireless Computing

The characteristics of mobile devices and the wireless environment that have been discussed above call for an architecture that reduces the load on the wireless link and support disconnected operations. We believe that mobile software agents are suitable to be used in a wireless environment for personalizing mobile services. A mobile agent is software entity that can move in a heterogeneous network to carry out various tasks on behalf of its user. It carries its actions in an autonomous, active, and reactive manner. Mobile agents have several advantages for mobile services [2]. Mobile agents run autonomously and asynchronously; mobile agents can adapt dynamically according to the execution environment; mobile agents can reduce network traffic; and they can be packaged with the specific tasks and sent to the destination host. When they arrive at their destination, they can carry out the tasks locally. It is also beneficial to reduce the flow of data in the network, especially when we need to handle a large amount of data stored in the remote hosts. Mobile agents are not without hurdles, however. The biggest hurdle is security. When a mobile agent moves to another platform to fulfill some tasks, malicious hosts may attack it. The security problems can be classified into three categories: (1) Security problems faced by agents; (2) Security problems faced by users; and (3) Security problems faced by hosts. Security issues of mobile agents are beyond the scope of this paper, but more information can be found in [3].

RELATED WORK

Several systems for personalizing mobile services have been proposed. An agent-based intelligent recommending system that can choose customized information for mobile users is presented in [4]. By learning the interests of users, this system can predict and select the information for mobile users. It includes five agents: a film agent, a news agent, a learning agent, a collaboration agent and an adaptation agent. Their functions are as follows: the film agent is in charge of collecting user's interests through different ways; the news agent is used to gather Internet news for user's identification; the learning agent can analyze the user's preferences; the collaboration agent is used to find other users with similar interests when making a decision; the adaptation agent can adapt the user's latest habits to improve the performance of the whole system.

In [5], a distributed personalization system that supports user mobility and allows multiple content providers to seamlessly join in is proposed. When providing personalized services to mobile users, this system considers not only the user's profile but also the characteristic of his device. It focuses on the wireless Internet in general, and does not depend on the specific wireless protocols such as WAP, and has no support for J2ME-based devices.

Other personalized mobile service platforms and frameworks are reported in [6] [7] [8]. They all provide personalized mobile services, but they are lacking in several ways. Firstly, they all provide mobile services based on WAP technology, none of them support J2ME-enabled devices. Secondly, they did not address the security issues for mobile agents in their systems. Thirdly, user preferences and profiles are not based on any standards such as CC/PP [1]. In order to design a robust, efficient personalized mobile service system [10], these issues should be considered, and that is what differentiates our proposed system. Firstly, our system can provide context-aware personalized mobile services to both, WAP and J2ME devices. Secondly, we use the CC/PP framework to design user profiles in our system. Through CC/PP framework, mobile customers can express their interests, preferences and device characteristics when requesting services, and the system can perform searches for services based on the information contained in the profiles. Thirdly, we provide a security module for agents and servers in the system. When mobile agents move to a server to perform tasks, mobile agents and the server authenticate each other. Through this method, we can prevent attacks from malicious agents or malicious servers [11].

THE PROPOSED AGENT-BASED SYSTEM

In this section we present the proposed agent-based context-aware personalized system for mobile services based on mobile agents. A high-level view of the architecture of this system is depicted in Figure 1. This architecture can be divided into three parts: server-side, client-side, and Internet-side (where services are located). The focus of our design is the server-side that

includes a WAP Gateway, Entry Server, and Service Server. The Entry Server accepts requests from mobile customers, creates personalized mobile services according to customers' context situation, and delivers these services to customers. The functionality of the Service Server is to provide mobile services for the Entry Server, so it often needs to update mobile services information from Content Providers on remote servers. The system can provide mobile services to J2ME-enabled as well as WAP-enabled devices. In order to satisfy this requirement, WAP Gateway was added in the system architecture. The WAP Gateway is responsible for transferring customer's requests and service results between customer's WAP devices and the Entry Server. On the client-side, customers can use both J2ME and WAP devices.

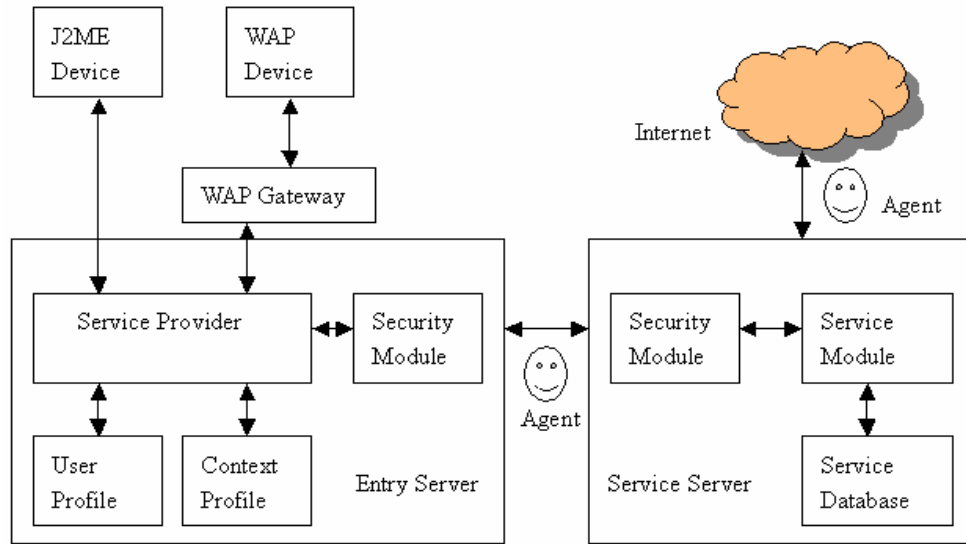


Figure 1: High-level architecture of the proposed system

Component Functionality

In this section we will discuss the functionality of the various components of the proposed system architecture. These components collaborate to deliver context-aware personalized mobile services to registered customers.

The Service Provider provides personalized mobile services to customers. When a mobile customer uses the system for the first time, the Service Provider will create a user profile and a context profile for the user, including preferences, device characteristics, and context information. After that, when this customer uses the system at a later time, the Service Provider authenticates the user using username/password. When a service request is received from a mobile customer, the Service Provider will check this customer's user profile to personalize her requests. Then, the Service Provider will send a mobile agent to Service Server to look for mobile services. When the mobile agent comes back with the results, the Service Provider will combine this customer's user profile and context profile to produce the personalized context-aware mobile services and adapt the contents to customers' current device interface. Finally, the Service Provider pushes the personalized content or services to the customer's mobile device.

The User Profile Module is used to store customer's user information in the system. Each customer's user profile describes the user's preferences and the characteristics of the device. In the proposed system, we use CC/PP framework to create user profiles for customers, and we will discuss this technology in detail later. If a mobile customer owns different handheld devices, the Service Provider will create different user profiles for her. Thus, the Service Provider can easily adapt the personalized content or services to the customers' specific device interface according to the current profile.

The Context Profile Module is also an important component in the Entry Server, and it is used to record customer's context information. In order to provide high quality personalized mobile services to customers, the system should consider customer's current context, such as location, time, etc. In our system, the Service Provider can track customer's context situation and store them in the customer's context profile. When the Service Provider needs to deliver personalized mobile services for a customer, it will filter some irrelevant information based on the customer's context profile and thus provide her with the information she is interested in. For example, some query results may return a long list of items, such as all

restaurants or theatres in a city. But the customer may just be interested in some localized information, such as restaurant locations within 1kilometer radius.

In any mobile agent system, security is very important. In order to enhance the security of our system, we add a **Security Module** in every Server. When a mobile agent moves to another Server, the agent and Server needs to authenticate each other. To enable this, we use Public Key Cryptography.

The Service Module is located in the Service Server, and it provides some remote service interfaces to the Entry Server. When the Service Provider needs to discover mobile services for a customer, it will send a mobile agent to the Service Server through the remote service interfaces. If the mobile agent can find the exact mobile services from the **Service Database**, it will return to the Entry Server along with the results; otherwise, the mobile agent will go to other remote servers to find mobile services needed by the customer. Through Java web service technology, the Service Module can also provide some interfaces for Content Providers to register their mobile services. Thus, the Service Module can always update the mobile services in the Service Database.

CC/PP FOR PERSONALIZATION

Web applications accessed from traditional HTML browsers running on desktops make assumptions about such client capabilities as screen size, bandwidth, support for color images, and so on. These assumptions break down when the same content is accessed from mobile devices, whose capabilities are more limited and varied. The challenge is how to support thousands of mobile devices with widely varying capabilities. Customizing content for different devices, and for different users, requires significant investments of time and effort. One way to ensure compatibility among the largest set of devices is to settle for the least common denominator, but then users of high-end devices are limited to the capabilities of lower-end devices, and there is little scope for user preferences. The challenge is then: how do you deliver content that reflects users' preferences and the capabilities of their devices without the time and effort of tailoring the code to each platform?

The World Wide Web Consortium (W3C) has finalized the Composite Capabilities/Preference Profiles (CC/PP), which is a proposed industry standard for defining user preferences profiles and describing delivery context. A CC/PP profile is a description of device capabilities and user preferences that can be used to guide the adaptation of content presented to that device, and it is constructed as a two level hierarchy: a number of components and each component has a number of attributes. These components and attributes describe the characteristics of a device, for example, hardware platform, screen size, and color capability. Industry leaders such as RIM, Nokia, Motorola and others have already begun to embrace this technology and have published profiles to support their devices. CC/PP will help optimize the content for a device, reduce the testing time, and even help create future-proof applications. An example vocabulary written for CC/PP is the UAProf.

Through CC/PP and other tools, the Service Provider can get mobile customer's preferences and device capabilities easily. On the client-side, the mobile device sends reference profiles and profile differences through some protocols such as W-HTTP or CC/PP-ex. A reference profile is usually presented as a URI, such as <http://nds.nokia.com/uaprof/N3650r100.xml>, and profile differences are sent as in-line XML fragments. When the reference profiles and profile differences are received, the Service Provider will use this URI to retrieve the default profile from a third source such as the website of a device vendor. The Service Provider can then merge and parse the default profile and profile differences through some tools, such as JSR 188 API, a standard API libraries provided by Sun Microsystems for processing CC/PP information in Java. Finally, the Service Provider can get the attributes and components of this CC/PP profile, and store them as a user profile in a database for future use.

IMPLEMENTATION

A proof of concept implementation has been constructed using Java technologies. The agent platform is implemented using Java Remote Method Invocation (RMI). In the Entry Server, the Service Provider is implemented as a Java Servlet, and it is in charge of communicating with mobile users and WAP Gateway. Our system supports three protocols developed for CC/PP, CC/PP-ex, W-HTTP (that use the HTTP protocol), and WSP (that uses the WAP protocol). When a customer's service request is received, the Service Provider creates the customer's user profile through JSR 188 API and stores it in User Profile Module. Customers can be tracked at a service point using wireless technologies, so a number of Entry Servers were provided as access points in service areas. When a mobile customer logs into the system, her device will try to connect to the nearest Entry Server. The Entry Server detects customer's context situation, such as location, and stores them in the Context Profile Module. In order to manage the user profile and context profile, a MySQL database is used in the Entry Server. The user profile and context profile are represented as tables in the MySQL database, and the Service Provider communicates with them through Java JDBC. The security module is implemented using Java Cryptographic Extension (JCE).

On the Service Server, we also install an MySQL database to store registered mobile services. Through Java web service technology, the Service Module receives mobile services information (or advertisements) from other Content Providers on the Internet and stores them in the Service Database. The Service Module provides some remote service interfaces to the Entry Server. The Entry Server sends our mobile agents to search mobile services from the Service Server or the Internet.

On client-side, whether the mobile customer uses a device based on J2ME or WAP, she can get mobile services from our system. The WAP Gateway can convert requests and services for WAP devices automatically. When a mobile user logs into our system, the Entry Server displays the service menu that the user can select services from.

OPERATING SCENARIO

In this section, we present two operating scenarios to describe how our system can provide context-aware personalized mobile services to users. Melissa and Jack are users of our system and they are interested in finding a mortgage calculator service to use from their mobile devices. Melissa uses a cell phone supporting J2ME, and Jack is using a WAP device. When Melissa logs in, the cell phone will send reference profiles and profile differences including CC/PP information to our system automatically. Once logged in, she will see the service menu on her cell phone as shown in Figure 2. Melissa selects the first service “Finance”, and another screen prompts her to enter a service name, so she enters “Mortgage” and sends out the request. On the server-side, when the Entry Server receives reference profiles and profiles differences, it will process them to create Melissa’s user profile and store it in User Profile Module. Through the user profile, the Entry Server can get some useful information about Melissa’s cell phone, such as ScreenSize, ColorCapable, etc. When the request is received, the Entry Server will check Melissa’s user profile. The Entry Server finds that the value of “JavaEnabled” attribute is YES, which means that the corresponding cell phone is J2ME-enabled, so it sends a mobile agent to the Service Server to find a mortgage service based on J2ME. Fortunately, the mobile agent finds the exact mortgage service in the Service Server, so it comes back along with the address of this mortgage service to the Entry Server. The Entry Server sends this address to Melissa’s cell phone. When received, Melissa presses a confirm key to accept it. The AMS (Application Management Software) of her cell phone will download and install the mortgage service. Finally, the service screen is displayed on Melissa’s cell phone and she can now start using the mortgage calculator as shown in Figure 3. Note that this J2ME application stays on the device until it is explicitly removed; therefore Melissa can use this service anytime she wishes. Also, it is important to note that Melissa doesn’t need to be connected to the wireless network in order to fully use the mortgage calculator.

Jack, on the other hand, is using a WAP-enabled device. When his request for a mortgage calculator is received, the Entry Server finds (through his profile) that his cell phone is WAP-enabled, so it sends a mobile agent to search for a mortgage service based on WAP. Unfortunately, the mobile agent did not find the exact mortgage service from the Service Database this time, so it goes to the Internet to search from other Content Providers. After searching, the mobile agent finds a website that can calculate mortgage payment online freely, so it returns to Entry Server the URL of the website offering this service. The Entry Server sends the URL to Jack’s device. If Jack accepts this service, his cell phone will try to connect to the website and the mortgage service will get displayed on his cell phone. A sample output is shown in Figure 4. Note that in the case of WAP, a service such as the mortgage calculator can perform the calculations on the device (if some sort of a scripting language such as WMLScript is used) or remotely in which case the user must be connected to the network. Also, it is worth noting that unlike J2ME services, WAP services are not installed on the device and therefore the user must download them every time he or she wishes to use them. If a device has support for both, WAP and J2ME, the user can then choose which service he or she wish to use, and such preferences can be entered into the preferences profile.

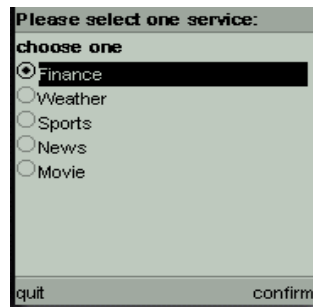


Figure 2: Service menu

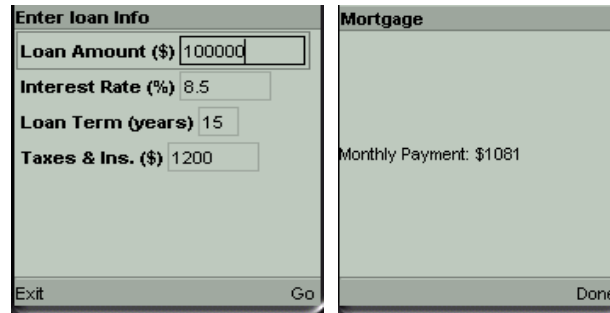


Figure 3: J2ME –based mortgage service

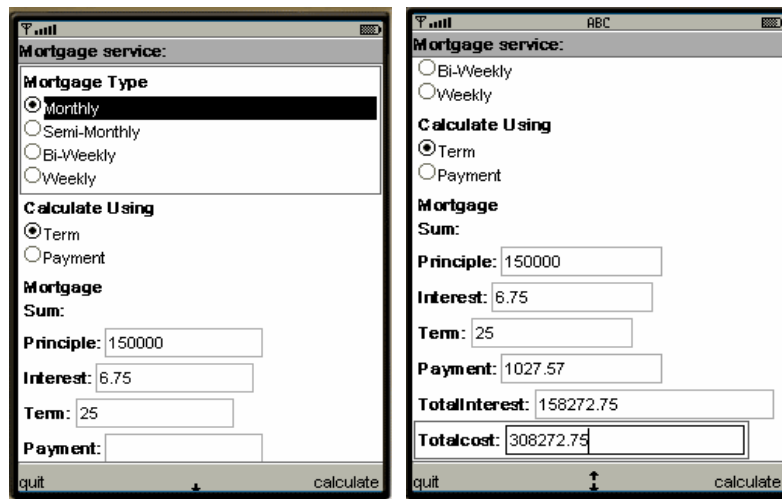


Figure 4: WAP-based mortgage service

CONCLUSION AND FUTURE WORK

In this paper, we have presented a mobile agent-based system for personalizing mobile services. We presented the architecture of the proposed system and discussed the functionality of its various components, and we described the proof of concept implementation we have built using Java technologies. Our approach is different from all related work in two main concepts: (1) we use a standardized profile representation known as the Composite Capabilities/Preferences Profile (CC/PP); and (2) we designed the system so that it supports two widely used technologies for developing mobile services, namely WAP and J2ME. Finally, we provided operating scenarios to demonstrate how the system can be used.

Work is already in progress to add localized wireless advertisements to the system so that value-added services can be offered to interested subscribers. For future work, we plan to add more services to the system and to enhance its functionality and evaluate its performance.

ACKNOWLEDGMENT

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant No. 045635.

REFERENCES

1. W3C CC/PP Information Page, <http://www.w3.org/Mobile/CCPP>.
2. Lang, D.B. and Oshima, M. (1999), "Seven Good Reasons for Mobile Agents," *Comm. of the ACM*, vol. 42, no. 3, pp. 88-89.

3. Claessens, J. Preneel, B. and Vandewalle, J. (2003) "How Can Mobile Agents Do Secure Electronic Transactions on Untrusted Hosts? A Survey of Security Issues and the Current Solutions," *ACM Transactions on Internet Technology*, vol. 3, no. 1, pp 28-48.
4. Lee, W.P. and Lu, C.C. (2003) "Customizing WAP-based Information Services on Mobile Networks," *Personal Ubiquitous Computing*, pp. 321-330.
5. Samaras, G. and Panayiotou, C. (2002) "Personalized Portals for Wireless User Based on Mobile Agents," *WMC'02, September 28, 2002, Atlanta, Georgia, USA*.
6. Wohltorf, J. Cisse, R. Rieger, A. and Scheunemann, H. (2003) "Berlin Tainment: An Agent-Based Serviceware Framework for Context-Aware Services," *DAI-Labor, Technische University Berlin, Salzuffer 12, 10587 Berlin, Germany*.
7. Jain, R. Puglia, S. Wullert, J. Parmeswaran, K. and Bakker, J. (2004) "The Mobile Application Server (MAS): An Infrastructure Platform for Mobile Wireless Services," *Information Systems Frontiers 6:1*, pp. 23-24.
8. Li, C.L. Lu, Z.D. and Li, L.Y. (2003) "Design and Implementation of a Hybrid Agent Platform," *Programming and Computer Software*, vol. 29, no. 1, pp. 28-42.
9. Sun Microsystems J2ME: <http://java.sun.com/j2me>.
10. Schoeman, M. and Cloete, E. (2003) "Architectural Components for the Efficient Design of Mobile Agent Systems," *Proceedings of SAICSIT 2003*, pp 48-58.
11. Bierman, E. and Cloete, E. (2002) "Classification of Malicious Host Threats in Mobile Agent Computing," *Proceedings of SAICSIT 2002*, pp 141-148.
12. Over The Air User Initiated Provisioning Recommended Practice Version 1.0, <http://java.sun.com/products/midp/OTAProvisioning-1.0.pdf>.
13. Farhoomand, A.F. and Drury, D.H. (2002) "Managerial Information Overload," *Communications of the ACM*, vol. 45, no. 10.
14. Personalization consortium, <http://www.personalization.org>.
15. Kendall, J.E. and Kendall, K.E. (1999) "INFORMATION DELIEVERY SYSTEM: An Exploration of Web Pull and Push Technologies," *Communications of the Association for Information Systems*, vol. 1, paper 14.
16. Hong, J.I. and Landay, J.A. (2004) "An Architecture for Privacy-Sensitive Ubiquitous Computing," *Mobisys'04, June 6-9, 2004, Boston, Massachusetts, USA*.
17. The WAP Forum (consolidated into the Open Mobile Alliance): <http://www.openmobilealliance.org>.