# Scenarios of using Web Services in M-Commerce

T. PILIOURA, A. TSALGATIDOU, S. HADJIEFTHYMIADES
Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens, Greece

The web service paradigm is a promising technology for developing applications in open, distributed and heterogeneous environments. The proliferation of this new technology has coincided with significant advances in the hardware and software capabilities of mobile devices. Due to the great benefits that come with the web service technology, such as interoperability, dynamic service discovery and reusability, there is a strong interest in making mobile devices capable of providing and consuming web services over wireless networks. This paper describes several scenarios of using web services in mobile devices and identifies their advantages, issues and challenges.

Categories and Subject Descriptors: D.2 **[Software]**: Software Engineering, K.4.4 **[Computers and Society]**: Electronic Commerce
Additional Key Words and Phrases: m-commerce, web services, wireless devices

## 1. INTRODUCTION

Web services (WS) constitute a new model for using the web that allows the web publishing of business functions and their universal access. Both developers and end-users can enjoy the benefits accruing from WS, as on one hand business application development and interoperation is simplified, and, on the other, end-user needs are well served through the provision of an intuitive, browser-based interface that enables them to choose, configure and assemble their own WS. At the same time, advances in mobile communications and the continuously increasing number of users of hand-held terminals make the latter an ideal channel for offering personalized information and services to mobile users and give pace to the rapid development of m-commerce, see for example the work in [8].

Given these advances in the technology landscape, it seems that the use of WS in mobile devices opens the possibility to provide services on the spot, no matter where the service requestor or provider is located. The dynamic discovery and invocation of services that the WS technology enables resonate with visions for mobile applications,

Authors' addresses: Department of Informatics and Telecommunications, University of Athens, Panepistimiopolis, TYPA Buildings, Ilisia, 157 84, Athens, Greece. Email: {thomi, afrodite, shadj}@di.uoa.gr

where the user's context may change dynamically, making different services[1] or service implementations appropriate at different times.

In the following we describe the WS model and related technology and then, we investigate two main scenarios of using WS in m-commerce and discuss related issues and challenges.

## 2. THE WEB SERVICE MODEL & RELATED TECHNOLOGY

### 2.1 The web service (WS) model and related standards

The WS model requires at least the following basic activities: describe, publish/unpublish/update, discover and invoke/bind, and contains three main roles: WS provider, WS requestor and WS broker. A WS Provider is the party that provides software applications for specific needs as web services. A WS Requestor has a need, which can be fulfilled by a service available on the Internet. A Requestor could be an application program, another WS or a human user accessing the service through a desktop or a wireless browser. A WS Broker is the party that provides a searchable repository of service descriptions where service providers publish their services and service requestors find services and obtain binding information for these services.

The use of standard technologies for service description, communication and data formats is essential for supporting the interaction between the above three parties. The most widely used standards for describing, advertising, discovering and binding WS in a distributed service-oriented environment are currently WSDL (Web Service Description Language) [17], UDDI (Universal Description, Discovery, Integration) [15] and SOAP (Simple Object Access Protocol) [12]. There are also several other approaches such as ebXML [4] and HP Web Services Platform [5]. A brief overview of the various approaches can be found in [14].

Despite the support offered by the various standards, there are a lot of issues related to WS description, composition, discovery, monitoring etc., which still remain open and are discussed in more detail in [14]. Currently, there is an increased interest in the use of WS in the mobile Internet world. In the following we describe existing and ongoing work on WS for Mobile Devices.

[1] The terms ‘web service’, ‘service’ and ‘WS’ refer to the same concept and are used interchangeably in the text

### 2.2 Using Web Services in Mobile Devices

When considering the world of mobile, internet-powered devices, the technical challenges associated with the development of the WS framework need to be extended to encompass issues like performance (efficient operation under restricted bandwidth conditions), cost and temporary unavailability (due to link outages). Some of the protocols used in the WS world of fixed devices are totally inappropriate for use in the expensive wireless medium. Apart from the bandwidth issue, wireless devices have also other handicaps like low CPU and memory capacities that need to be taken into account.

Key commercial players like Microsoft and SUN recognize the importance of extending the WS paradigm into the wireless world and introduce specific features in their products. Microsoft has released the .NET Compact Framework Beta in early 2002. SUN plans to extend their J2ME platform to fully support WS. This is mostly pursued through the JSR (Java Specification Request) No 172.

## 3.SCENARIOS OF USING WEB SERVICES IN MOBILE DEVICES

In this section we examine two scenarios for using WS in mobile devices and we discuss related work, associated issues and challenges. In the first scenario the mobile device assumes the role of the WS requestor under two possible architectural configurations. The second scenario involves a totally different configuration whereby the mobile device plays the role of the WS provider. We should stress out that a number of combinations of these scenarios are also encountered in today's mobile applications but they are not mentioned here, since they introduce no extra technical challenges to the ones discussed below.

### 3.1 Mobile Device acting as WS requestor

This scenario is best suitable for nomadic users who wish, for example, to locate a product or service close to their physical location, to manage their personal information that is hosted on a central server and to administer their systems by invoking relevant WS. Furthermore, this scenario enables mobile users to directly participate to the next development stage of the Web, referred as Semantic Web [1]. In [10], a prototype Semantic Web environment for context-aware mobile services is described. These mobile services aim at enhancing everyday campus life. They combine location, calendar, weather and social context information along with personal preferences and permission profiles, in order to provide timely and relevant information and services to mobile users.

However, in order to enjoy the full benefits of this scenario, a number of challenges must be met. Some of these challenges are specific to an architectural configuration (and thus described in the relevant sections) and some others are more generic, such as automated WS discovery/execution and WS personalization. The WS discovery and execution need to be automated to bring the ultimate ease-of-use to the mobile users. Furthermore, for WS to better serve mobile users' needs they need to be personalized according to users' profiles, needs and contexts. This requires describing the semantics of WS and also capturing the context within which the user operates. An infrastructure to make semantically described WS available to mobile devices through agent technologies and service registries is described in [3].

### *3.1.1 WS-aware mobile device*

In this architectural variation of the first scenario, the entity that plays the role of the WS requestor is the mobile device itself (see Figure 1). This mobile device needs to dispose a WS client application in order to enable the provision of services to mobile users. It interacts with the service provider and the service broker using WS-aware protocols over the wireless network (e.g., WLAN, GSM/GPRS). This means that the mobile device is a *fat client* with XML processing capabilities. Fat clients provide more sophisticated support in terms of application specific functionality [2]. Using WS in building these fat clients have numerous advantages: (i) it enables handset manufacturers to rapidly deploy Internet solutions built on global standards, (ii) it makes applications more dynamic as they can invoke different services or service implementations based on the user's context and (iii) it facilitates the interoperability and integration with enterprise applications and with applications running on other mobile devices.

This scenario is particularly applicable in cases where an interaction with multiple companies is needed. This interaction must be fully automated without user human involvement. For example, in case the mobile user is looking for a hotel, the WS client contacts the WS broker in order to find the closest hotel, then it interacts directly with the service provided by the hotel in order to negotiate the price and finally it makes the reservation by paying a bond using the mobile user's credit card.

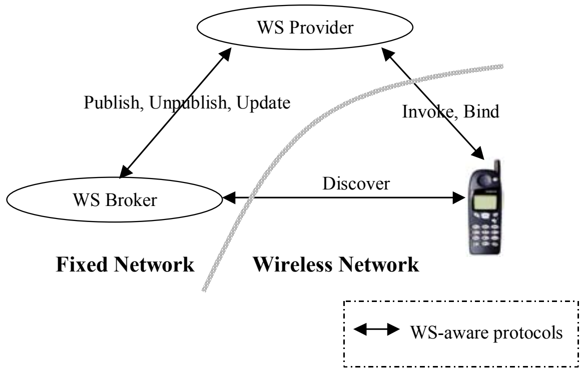


Fig. 1. Mobile phone as a service requestor

There are several issues and challenges related to this scenario which come from the fact that mobile devices have much lower processor power, limited bandwidth, less memory, limited display, restricted input devices and a finite battery power when compared to desktops. Furthermore, depending on the resource capacity of the mobile device, different markup languages or output formats have to be used. For example, a PDA running Palm OS can present data as TinyXML [13], while a device capable of interpreting VoiceXML [16] can allow the user to listen to data instead of viewing it. Therefore, mobile services need to deal with device limitations and characteristics and be optimized to run on these devices. For example, due to the difficulty in using the keypad, well-designed services should require minimal user input. In [2] the issue of limited bandwidth is partially tackled by a two-phase extension to the SOAP protocol.

It seems that the particularities and limitations of mobile devices pose considerable barriers to the adoption of this scenario. For example, when using UDDI for service discovery multiple costly network round trips are needed. This is troublesome in mobile networks, as the unavailability of the network may hinder the completion of the user request. The aforementioned problems lead us to the second architectural configuration described in Section 3.1.2.

### *3.1.2 WS-agnostic mobile device – A proxy-based architecture*

In this architectural variation, we introduce a proxy entity that plays the role of the mobile device representative (see Figure 2) in the fixed network infrastructure. This proxy interacts via WS-aware protocols with the service broker and the service provider and returns the results to the mobile device using WS-agnostic protocols (such as

WAP/WML, i-Mode/cHTML) over a wireless network. Its role is the provision of services in a way adapted to the capabilities and the constraints of a mobile device (see previous Section). The mobile device doesn't need to support WS functionality (*thin client* scenario) as the workload is passed to the proxy thus relieving the mobile device of the time- and processor-consuming WS related tasks. The functionality of the proposed proxy is quite similar to that of the WAP gateway (used by contemporary mobile operators) performing adaptation between conventional WWW protocols (i.e., HTTP/TCP/IP) to wireless-optimised variants (i.e., WSP/WTP/UDP).

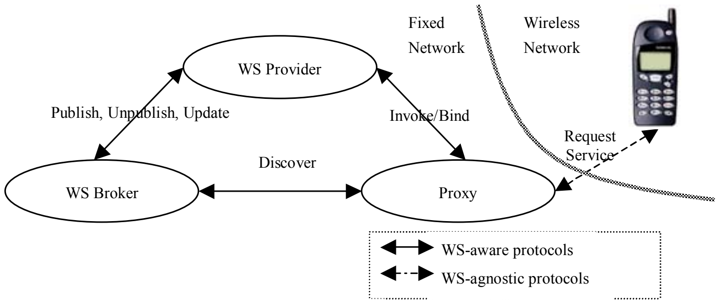


Fig. 2. Proxy as a mobile phone representative

The proxy could be used to handle the fact that mobile devices non-deterministically lose network connectivity much more than wired applications. The proxy can keep the results of the service invocation and forward them to the mobile device when the connectivity is re-established. The proxy may also perform various tasks such as conversion and content adaptation in order to adjust the WS result to different terminal and network environments. For instance, a proxy server may minimize the information flow over low/medium speed wireless links. Furthermore, the proxy can take care of various housekeeping chores for the mobile devices, such as keeping the mobile user's bookmarks, managing its cache and so on, to relieve the memory and power limitations of the mobile device. This scenario is particularly applicable in the case where the mobile user moves into an unfamiliar environment and obtain services for which it has no previous knowledge. For example, we could consider a mobile user entering an airport and obtaining access to services such as flight information, special offers and promotions in the duty free shops, etc.

However, there are also some challenges related to this variation. The proxy is a central entity that controls all user data and decides what services are accessible by the mobile device. This can be extended to a wireless portal network business model that materializes the so-called “walled garden” [6]. This model could cause serious business concerns in case that the proxy is controlled by an untrustworthy, insecure or abusive monopoly. Furthermore, the proxy constitutes a single point of failure, which exposes the whole network to attacks.

### 3.2 Mobile Device acting as WS provider

A challenging scenario regarding the provision of WS is the one involving the WS service provision at a mobile terminal. Nowadays, the capabilities offered by portable equipment are getting more and more advanced. For example, we have seen installations of Web/application servers and servlet engines in PDAs, delivering static and dynamic content to fixed and mobile users of the network infrastructure. Hence, the case of a mobile device hosted WS provider (hereinafter to be referred to through its acronym MH-WSP) scenario is highly foreseeable and technically feasible. This scenario greatly facilitates the provision of WS that are context sensitive and, in particular, location based services (LBS) [7]. This family of value-added services experiences a vast penetration and wide acceptance in today's telecommunication organisations and it is also anticipated that they will assume a vital role in the upcoming rollout of 3G and 4G networks (e.g., GPRS, UMTS). The development of the LBS world is equally important to the introduction of the so-called pervasive computing framework [11].

We could consider various applications where this scenario can be used. For example, the mobile devices could contain all personal information such as social security number, medical background and curriculum vitae as well as services for accessing this information. Then, service requestors, such as agents sent by employers, could invoke these services in order to find, for example, potential employees. This scenario could also be used in fleet management applications where the mobile device could act as the provider of a service, which denotes the actual location of a specific truck.

A number of issues and technical challenges are associated with this scenario. Since the WS framework relies on the use of registry services like most distributed systems do, the mobility of the WS provider needs to be handled somehow. Typically, when internet-powered mobile nodes move to another location, a new address is obtained. Such address may be somehow associated with a well-known address (Mobile IP paradigm) or be totally different (e.g., dynamically assigned). Mechanisms like NAT (address translation) make this case even more complicated. By changing its address, the MH-WSP becomes partly inaccessible by the rest of the world. Provision should be taken to update the WS registry on the new address where services will continue to be offered to their requestors, much alike the mobile IP mechanism (Fig. 3).

MH-WSP could rely on currently available APIs for automatically updating the UDDI service when relocations are experienced. Such functionality is quite critical and should, potentially, be incorporated in the emerging mobile WS frameworks where issues like efficiency in portable devices are mostly considered at the moment.

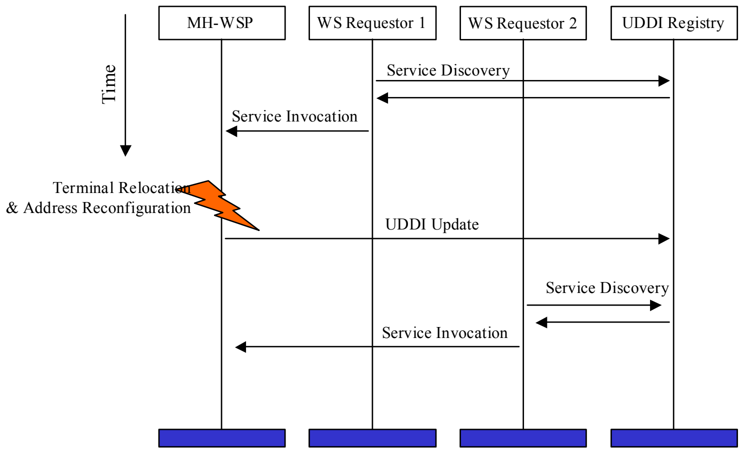


Fig. 3. UDDI Update After Mobile Device Address Reconfiguration

Another option regarding the seamless provision of the service in the wireless/mobile world is to hold the WS provider responsible for immediate notifications to known clients-requestors following the occurrence of a terminal relocation and related side-effects. This applies in cases where the number of requestors is limited and there is no entity acting as a WS broker. Provisions for such notification protocol are not taken in the current standards. Each WS provider should maintain a table with the addresses of known clients. Through such a table, clients are notified on the provider's relocation and change of address.

Lastly, the registry update/notification operation should be protected against malicious use. Hence, the operation presents a potential security problem due to the change of terminal's point of attachment. Contemporary security mechanisms may react to such a change and reject the update request. Therefore, it is extremely important to resort to mobility – aware security frameworks. Indicative example of such a framework is the Wireless Application Protocol PKI, introduced to address widely distributed personal trusted devices (PTDs) that support the realization of financial transactions and other activities related to m-commerce[17]. WPKI has been specified by OMA – Open Mobile Alliance [9] (previously WAP Forum).

## 4. CONCLUSIONS

The use of WS on mobile devices will augment traditional voice communication with mobile services, which is making possible the access to any data and service, on any device, any time. In this paper we have presented some scenarios that make this possible. Out of the three presented scenarios, only the first and second ones are feasible nowadays but not necessarily efficient. This is reconfirmed by some prototype developments we have attempted. In order to enjoy the real value of WS and mobile devices, the third scenario requires to also be incarnated. Since most of the technologies supporting these scenarios are still or just emerging, many challenges prevail, such as performance, security and user interface issues and need to be further investigated.

## REFERENCES


[1] BERNERS-LEE, T., HENDLER, J., LASSILA, O. 2001. The Semantic Web. In *Scientific American*, May 2001, available at http://www.sciam.com/issue.cfm?issueDate=May-01

[2] DAHLEM, D., Jahnke, J. H., ONABAJO, A., AND Wang, O. 2002. The Challenges of Implementing Web Services on Small Devices. In *Proceedings of the Workshop on Pervasive Computing, Going Beyond Internet for Small Screens, OOPSLA 2002*, Washington, USA, November 4th, 2002.

[3] DOGAC, A., LALECI, G., KURT, G., KABAK, Y., AND ACAR, A. 2002. A Platform for Semantically Enriched Mobile Services. In *Procs. of the First International Conference on Mobile Business*, Athens, Greece, July 2002.

[4] ebXML, http://www.ebxml.org

[5] HP Web Services Platform, http://www.hpmiddleware.com/products/hp_web_services/

[6] KALAKOTA R., AND ROBINSON M., 2001. *M-Business: The Race to Mobility*, McGraw Hill.

[7] Location Inter-operability Forum (LIF). Mobile Location Protocol. LIF TS 101 v2.0.0, November 2001.

[8] MAAMAR, Z., MANSOOR, W., YAHYAOUI, H. 2001. E-Commerce through Wireless Devices. In *Proceedings of IEEE 10th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'01),* IEEE Press, 328-333.

[9] OMA, http://www.openmobilealliance.org/

[10] SADEH, N. A Semantic Web Environment for Context-Aware Mobile Services. http://www.wirelessworld-research.org/BoV1.0/Contributions%2003/03_2_02_Norman_M Sadeh _v2.pdf

[11] SATYANARAYANAN, M., 2001. Pervasive Computing, Vision and Challenges, *IEEE Personal Communications*, Vol. 8, No.4.

[12] SOAP, http://www.w3.org/TR/SOAP/

[13] TinyXML, http://www.gibaradunn.srac.org/tiny/

[14] TSALGATIDOU, A., T. PILIOURA, 2002. An Overview of Standards and Related Technology in Web Services. In *International Journal of Distributed and Parallel Databases, Special Issue on E-Services*, September 2002, Vol. 12, No. 2, 135-162.

[15] UDDI, http://www.uddi.org

[16] VoiceXML, http://www.w3.org/TR/voicexml20/

[17] WSDL, http://www.w3.org/TR/wsdl

[18] YEUN, C. Y. AND FARNHAM, T. 2001. Secure M-Commerce with WPKI. In *Proceedings of 1st International Workshop for Asian PKI*, Korea, October 2001.