
Providing web services to mobile users: the architecture design of an m-service portal

Minder Chen*

Decision Sciences & MIS, School of Management,
George Mason University, Fairfax, VA 22030, USA

E-mail: mchen@gmu.edu

*Corresponding author

Dongsong Zhang and Lina Zhou

Department of Information Systems, University of Maryland,
Baltimore County, Baltimore, MD 21250, USA

E-mail: zhangd@umbc.edu

E-mail: zhoul@umbc.edu

Abstract: In order to reduce costs and gain competitive advantages, many companies are offering web services that enable software components to be accessed by their customers and trading partners over the internet. With the latest advances in mobile computing and communication technologies, web services are increasingly used to expose functionalities of many legacy systems so that more business-oriented mobile applications can be developed. In this paper, we first use a generic technical architecture. Then, we provide a classification matrix for grouping various mobile services. The architecture design of an m-service portal is proposed, followed by the description of an m-service prototype in support of e-procurement developed by the authors. Potential enhancement to the m-service portal using agent technology is also discussed. Finally, we suggest some further research issues of dynamic web services in the context of m-service portals and context-aware m-services.

Keywords: web services; semantic web; m-service; m-commerce; mobile computing; supply chain management; e-procurement.

Reference to this paper should be made as follows: Chen, M., Zhang, D. and Zhou, L. (2005) 'Providing web services to mobile users: the architecture design of an m-service portal', *Int. J. Mobile Communications*, Vol. 3, No. 1, pp.1–18.

Biographical notes: Minder Chen is an Associate Professor at the School of Management, George Mason University at Fairfax, VA. He received his PhD in Management Information Systems from the University of Arizona. His research interests include collaboration technologies, web services, mobile computing, and electronic commerce.

Dongsong Zhang is an Assistant Professor in the Department of Information Systems at University of Maryland, Baltimore County. He received his PhD in Management Information Systems from the University of Arizona. His research interests include web-based learning, mobile computing, web service applications, and computer-mediated communication.

Lina Zhou is an Assistant Professor in the Department of Information Systems at University of Maryland, Baltimore County. Her research interests center around text mining, deception detection, ontology learning, and semantic web.

1 Introduction

With the advent of service-centric computing and the widespread application of web technologies, the internet offers unprecedented opportunities to provide more integrated services and streamline supply chains. Many organisations have moved their business online in order to achieve competitive advantages by reducing cost, increasing customer satisfaction, enhancing business collaboration and partnership, and extending service offerings. The use of web services is increasing rapidly as the need for application-to-application communication and interoperability grows.

A web service, defined by W3C, is a software system identified by a URI (Uniform Resource Identifier) whose public interfaces and bindings are defined and described with XML. It is a stack of emerging standards that describe a service-oriented, component-based application architecture. The web service standards stack includes internet protocols for networking, XML for data representation format, SOAP (Simple Object Access Protocol) for messaging protocol, WSDL (Web Service Description Language) for service definition, and UDDI for service registration. Web services have become a critical enabling technology for system integration within an enterprise or among business partners.

The workforce in the 21st century becomes increasingly mobile. A recent IDC study predicts that the number of mobile workers in the US will rise from 92 million in 2001 to 105 million in 2006 [1]. By 2006, two-thirds of employees in the US will be mobile workers, who will spend more than 20% of their time away from their desks – attending meetings, traveling, or performing field works. Some studies indicate that mobile workers' productivity may be improved by 30% when proper mobile technologies are deployed [2]. With the latest advance in mobile computing and communication technologies, we are beginning to see a variety of wireless applications using web services (m-services).

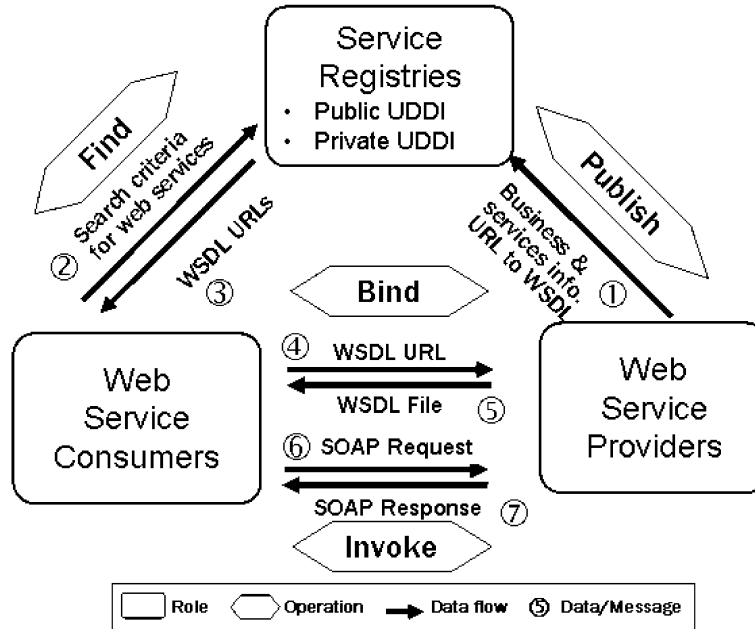
The term *m-service* extends the concept of web services to the wireless environment. It refers to requesting and running web services on wireless devices. An application component is considered as an m-service if it is [3]: transportable through wireless networks; flexible in terms of composition with other m-services; adaptable according to wireless devices' computing characteristics; and accessible by wireless devices via a micro browser as a mobile web application or able to interact with a mobile client software component that consumes web services. Examples of m-services include context-aware mobile applications such as location-based advertising and tourist services, financial applications such as mobile banking, and mobile e-commerce such as inventory tracking and real-time trading. Compared with web services, m-services face more technical, legal, and organisational challenges due to various restrictions from both wireless devices and networks.

It has been well recognised that a major challenge to IT architects and business organisations is to articulate the potential impact of web services and m-services on the development of information systems, IT infrastructures, and business processes. In this paper, we first introduce the concepts of web services and m-services, as well as their relationships. A matrix is used to classify m-service applications. Then, we propose a technical architecture of an m-service portal. A prototype e-procurement m-service is developed to exemplify the architecture. Next, an intelligent m-service architecture is presented to enhance the service searching and matching process. Finally, we discuss a few related research issues such as semantic web services and context-aware applications.

2 Web services

Web services provide a standard-based approach to implementing distributed components. They offer data and business logic services over standard protocols such as HTTP, XML, and SOAP over the internet. Gartner defines web services as “loosely coupled software components delivered over internet-based technologies [4]”. The term ‘loosely coupled’ implies that web services are independent of any programming languages, platforms, and object models. Using the ubiquitous and low-cost internet, web services can easily provide interoperable software functions over the intranet and the internet.

Web services are a set of standards to form a service-oriented architecture (SOA), as depicted in Figure 1. This architecture models the interactions between three roles: service providers, service consumers, and service registries. The interactions involve service publishing, finding, and binding operations. Interfaces to a web service implementation are defined in the Web Services Description Language (WSDL). A web service provider can publish the URL of the WSDL file of a web service in a service registry, such as a public UDDI (Universal Description, Discovery, and Integration) business registry. A web service requester (consumer) can search and retrieve the service description and the access point to the WSDL file of a service from a service registry. The WSDL file will be used to generate a client-side web service proxy. Then, the service consumer uses service description to bind with the web service. At the runtime, it sends a SOAP message to the service provider to invoke a specific method of the web service and receives a SOAP response. A SOAP message is an XML-based construct that includes an optional SOAP header and a mandatory SOAP body. Using web service standards, software components can be accessed by applications developed by business partners regardless of hardware, operating systems, and programming languages used.

Figure 1 The service-oriented web services architecture

From a distributed computing perspective, web services consist of distributed and reusable software components that encapsulate discrete functionality semantically and are programmatically accessible via standard internet protocols. They can be used as a wrapper to expose functions of legacy systems to applications inside and outside enterprises, including mobile applications. Therefore, web services are viewed as a new middleware technology for building middle-tier software components that implement business functions and data access logics.

3 A technical infrastructure of m-services

In order to improve operational performance, enhance business processes, and increase productivity to remain competitive, mobile service providers are seeking cost-effective ways of integrating a variety of services into their mobile portals. The convergence of the internet and wireless technologies is enabling new services that people could never, until recently, have imagined. On July 17, 2000, the largest mobile phone operator in the USA, Verizon Wireless, launched an internet service called 'mobile web', in a bid to become the nation's largest wireless web service provider. It offers services including local information, search engine functions, shopping (through Amazon.com and Barnes & Noble), organiser functions, and e-mail service.

There are a number of emerging technologies and protocol enhancements designed to extend online services to users of mobile devices. New protocols such as WAP (Wireless Application Protocol) enable companies to design miniature websites specifically for mobile phones with small display screens and limited network bandwidth. These services need to be integrated as web services in order to provide interoperability, allowing users

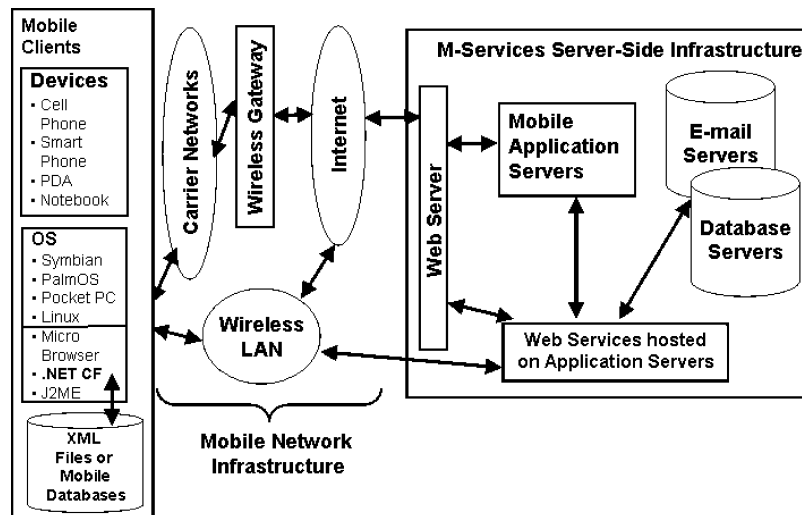
to link loosely coupled web service components to form composite services. The wireless application environment (WAE) supports interaction between wireless protocols (e.g., WAP and iMode), web applications, and wireless devices equipped with micro browsers capable of rendering web pages encoded in wireless markup languages such as WML and cHTML.

There are a variety of m-services applications. For example, a user may use a WAP-enabled phone to browse websites of movie theatres and order movie tickets. In the meanwhile, the web applications of movie theatres can deliver information about new releases or previews to mobile users through multimedia messaging service (MMS). With the widespread deployment of mobile devices and communication, mobile e-commerce (m-commerce) is becoming the next phase of electronic business. It is estimated by Gartner Group that at least 40% of B2C e-commerce will be initiated from smart phones supported by WAP in 2004. There has been some research on both theoretical and practical issues of m-commerce. For example, Varshney and Vetter [5] propose a framework for implementing mobile commerce applications. Other examples of m-services are mobile financial applications and context-aware tourist guide, where tourists carrying location-aware mobile devices can obtain relevant tour guide in real-time.

A number of benefits can be gained from m-services, including getting access to real-time information anytime and anywhere; providing faster responses to customer queries and needs, thus increasing customer satisfaction and reducing the cost; triggering business events and initiating business transactions from mobile devices to shorten the cycle time of core business processes; and enabling mobile access to obtain up-to-date intranet and extranet content and prevent data inconsistency.

Figure 2 shows a generic technical architecture of m-services based on the past work of IT vendors and researchers [6–8]. On the client-side, mobile devices can consume web services via the following ways:

Figure 2 A wireless environment for m-services



- *Mobile web clients.* PDAs or web-enabled cell phones can use mini web browsers to access mobile web applications via wireless internet connection [9]. Mobile web applications are hosted on application servers and can consume web services on the server-side. They can invoke web services, integrate responses from these web services, and then return the consolidated results as web pages to mobile devices.
- *Mobile local clients.* Similar to traditional client-server applications, mobile local client applications run on mobile devices and connect to server components via a wireless local area network (WLAN). After receiving service requests, web services can deliver content in appropriate formats to mobile devices running m-service clients (i.e., SOAP clients) that are capable of processing web service responses locally. Data retrieved from servers can be stored in a local database on the mobile device or as XML files. As a result, mobile users can still work with the data while working offline. The Java community and Microsoft have provided tools and platforms for building such kind of fully functional web service clients on mobile devices. Such mobile devices need to have J2ME or .NET Compact Framework installed in order to run SOAP clients to consume web services deployed on the server side.

On the server side, web services are usually deployed on application servers. They are used as business-logic-layer software components in a multi-tier software architecture to provide computation and data access functions. They can interact with other web services distributed across the internet, as well as server-side components implemented as classes (e.g., Enterprise Java Bean or Microsoft COM components) within the same enterprise network. Mobile application server component provides a broad range of functions including content adaptation and notification.

- *Content adaptation or transcoding services.* Mobile devices may support different markup languages. Creating content in different formats for various mobile devices is very costly and impractical. There are server-based solutions, such as IBM's WebSphere Transcoding Publisher and Oracle 9i Wireless, that can dynamically translate and optimise web content and applications into several appropriate markup languages before delivering them to mobile devices. Microsoft takes a different approach by providing a programming tool, mobile information toolkit, to support development of mobile web applications in one code base. Therefore, programmers do not have to create separate content in different markup languages for various mobile devices.
- *Notification services.* Notification services provide data subscription and delivery mechanisms. Applications that use notification services can provide users with the needed information any time, empowering them to make timely and informed decisions. Users subscribe to these services by specifying triggering events for generating and sending notifications. Notifications will be sent to users' mobile devices as SMS (Short Message Service) or to their email accounts.

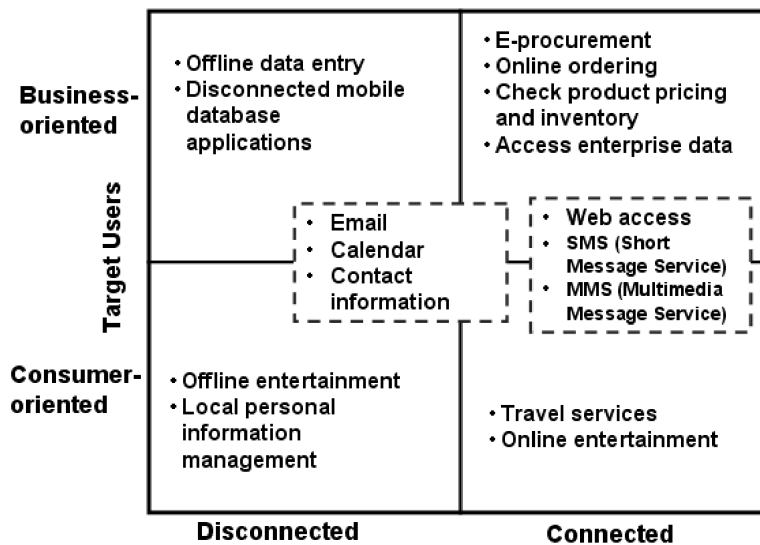
4 Classifications of m-services and business opportunities

In the context of m-services, web services can be viewed as an enabling technology that encapsulates business logics and data access mechanisms and exposes business functions to mobile clients. In this section, we will first present a classification matrix of m-services, and then discuss potential business opportunities.

4.1 Classification of m-services

M-services include services that can be consumed in either connected or disconnected mode by consumers or business applications [10]. Figure 3 presents a matrix that classifies m-services based on two dimensions: target users (consumers vs. business) and network connection (disconnected vs. connected). Disconnected applications can run in an offline mode. Data entered in the offline mode will be transferred and synchronised with the centrally stored data when mobile devices are connected to the network later on.

Figure 3 A classification of mobile services



E-mail service, calendar function, and contact information management serve as basic building blocks to m-services from all four quadrants in Figure 3. For example, e-mail is an essential function in a mobile environment because it is often used in the context of business workflows in support of mission-critical enterprise applications. Web access, SMS, and MMS are basic services that help deliver other value-added m-services in the connected mode. Business-oriented m-services in the connected mode include e-procurement, online price checking and ordering, shipping status tracking, and so on. These m-services can be used to integrate mobile applications with legacy systems or ERP systems.

4.2 *Business opportunities for m-services*

Mobile computing technology helps make the right information available to the right person for performing critical business activities regardless of time and location. Mobile applications can be developed in several strategic areas to support employees, customers, and trading partners. M-service applications directly used by employees include horizontal applications related to personal information management and e-mail service. Vertical mobile applications provide direct support for enterprise applications such as sales force automation, field work automation, inventory control, warehouse management, and logistics.

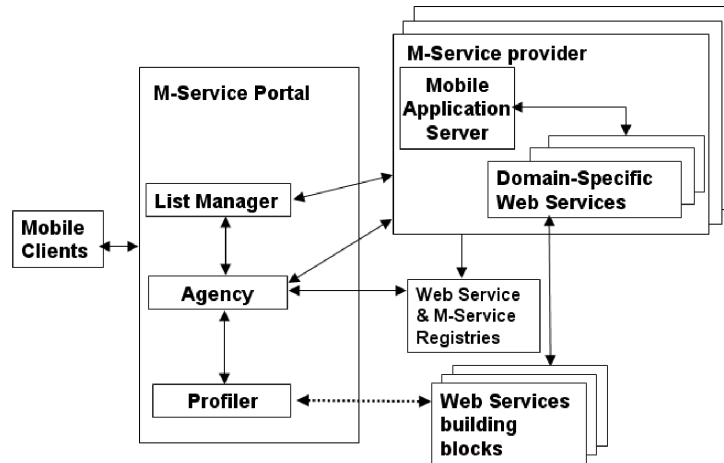
M-commerce offers many business opportunities [11]. Wireless content delivery service is a critical element of m-commerce. Popular content-oriented m-commerce applications are financial news delivery, stock portfolio tracking, stock quote checking, and telephone directory assistance. Marketing applications in m-commerce may automatically alert users of special sales at nearby shops based on their locations and interests. Transaction-oriented m-commerce such as trading stocks often requires immediate action even when people are traveling [12]. M-commerce applications usually include micro-payment mechanisms that allow mobile users to pay for products purchased from vending machines or to buy tickets through mobile devices.

Businesses should also be aware of the difficulties while using m-services [13]. M-services should be designed and implemented in such a manner that much fewer steps and data entry efforts are required to complete a transaction. Providing context-aware m-services may reduce data entry operations for the mobile user in areas such as supply chain management [14]. Using location-based m-services to advertise sales information to customers at the proximity of stores may attract more customers and increase the revenue [15,16]. Innovations in B2C mobile applications can be found in some vertical industries such as retailing [17]. Providing mobile web access for a B2B exchange, which allows users to submit bids through their mobile devices and receive alerts of new bids, seems to be a promising application.

5 An m-service portal

5.1 The architecture of an m-service portal

A web portal is an aggregation of heterogeneous web resources at a website to provide easy access to these resources. We propose an m-service portal architecture, as shown in Figure 4, which integrates m-services to provide adaptive and personalised services and to accommodate the constraints of mobile devices. The architecture can potentially support all types of m-services discussed in Section 4.

Figure 4 An m-service portal architecture

An m-service portal can be either consumer-oriented or business-oriented. The architecture of the proposed m-service portal consists of three major components:

- *List manager* maintains a personalised list of pre-configured m-services (e.g., frequently-used hyperlinks to various m-services) specified by mobile users. The list can be updated based on mobile users' access patterns by assigning higher scores to the most recently visited sites. It aims to reduce unnecessary data entry and search. The user can also use the List Manager to link to other m-service portals such as:
 - *Google Wireless WebSearch*. It allows users to search not only the 'Mobile web' created specifically for mobile devices, but also the entire World Wide Web.
 - *MSN Mobile*. It provides mobile internet services similar to MSN services to users of mobile devices. Those services include e-mail, weather information, sports, news, personalised MSN alerts, as well as MSN Messenger.
- *Profiler* is responsible for storing personal information and preferences such as financial profiles, health and insurance profiles, information needs (such as scores of sports games and local weather), message delivery preference for consumer-oriented m-services, and properties of mobile devices that users have (for customised content presentation) [18]. Business rules can be integrated into business-oriented m-services such as e-procurement. The management of profiles and preferences can be implemented using web services built upon external web service building blocks, so that the profiles and preferences can be easily shared with other m-service applications outside this m-service portal.
- *Agency* uses intelligent agent technologies to reduce unnecessary interaction between m-services and the mobile user. Based on users' information recorded in the Profiler, agents can proactively collect relevant services and information on behalf of users. When an m-service is not found in the List Manager, the agent(s) can automatically search in the web service/m-service registries to identify appropriate services that satisfy users' needs. The semantic web services technology [19] can be incorporated into the design and implementation of the Agency.

There are three other essential elements that interact with the m-service portal. The first one is a set of building-block web services. They are common basic web services required by most m-service applications. Some software vendors and third party developers (e.g., Microsoft and Oracle) have been developing 'foundation' web services as building blocks for complicated web service-based applications. Typical building blocks for m-services include the single sign-on service supporting user authentication, the alert service providing automatic message delivery, micro-billing services that track transactions made by mobile users, and the GIS (Geographic Information System) mapping service supporting the development of location-based services.

The second element is m-service registries that contain a list of services published by m-service providers. The UDDI standard can be used to implement m-service registries. Current public distributed UDDI Business Registry nodes are maintained by companies such as IBM and Microsoft. Services registered at different nodes are synchronised on a daily basis. Therefore, a web service provider only needs to publish their services in one of the distributed registries that are listed at www.UDDI.org. The UDDI standard is too broad in its definition of 'Services' and does not provide enough semantics to support effective m-service search. We envision that several vertical UDDI-based registries designed to support m-services in a variety of industries (e.g., travel, finance, and entertainment) will emerge to complement existing public UDDI registries.

The third element is a group of m-service providers that develop and host m-services particularly useful for mobile users. They are likely to deploy mobile application servers with transcoding services, which enable the information delivered by web services to be consumed by mobile devices' micro browsers directly instead of by local web service clients.

5.2 *A prototype of an e-procurement m-service portal*

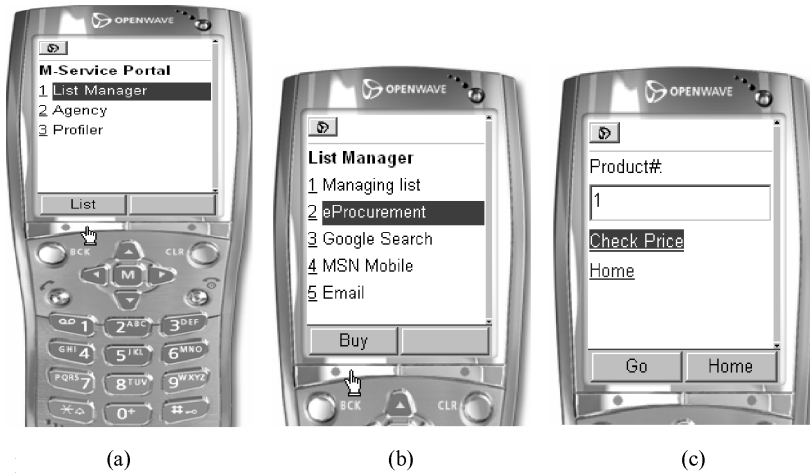
This section describes an e-procurement m-service prototype that we have developed in the context of the m-service portal. This m-service allows purchasing personnel to inquire product prices from suppliers and to place an order from a mobile device connected to the internet. This e-procurement m-service application allows employees to replenish an inventory while they are checking it in a warehouse.

In the application, each supplier listed in a private web service registry needs to implement two web services: QueryPrice (price checking) and ProductOrdering. Both web services can be implemented with any programming language and run on any platform. They are used by an e-procurement m-service for buying products from suppliers. The e-procurement m-service is built with ASP.NET and Microsoft Mobile Internet Toolkit (MMIT). MMIT supports the development of mobile web applications and provides a context adaptation function capable of rendering generated content dynamically in support of different standards such as WML and cHTML to mobile devices. This m-service prototype is developed in Visual Studio .NET and deployed in the .NET Framework environment.

Openwave Mobile Browser Simulator is used to test the prototype system. A play-by-play description of an e-procurement m-service scenario and operations performed behind the scene are described as follows. Figure 5(a) shows the homepage of the m-service portal with links to its three major functions. The mobile browser is used to present content that is dynamically generated and rendered in WML. A user selects the *List Manager* by pressing the button below a softkey labelled as '*List*'. A list of

frequently used m-services and a ‘*Managing list*’ link that allows user to configure the listing of m-services will then be displayed, as shown in Figure 5(b).

Figure 5 (a) The home page of the m-service prototype, (b) Selecting e-procurement from the list manager and (c) Entering a product ID to inquire prices from vendors via the *QueryPrice* web service



If a user selects the e-procurement m-service, he or she can enter a product ID and then choose *Check Price* link to activate the dynamic price checking function, as illustrated in Figure 5(c), to inquire the prices of this product from a number of qualified suppliers. We added a web reference to the URL of the WSDL file of the *QueryPrice* web service provided by a supplier and referred to it as *QueryPrice* locally. A client-side web service proxy is then generated. The price checking function (i.e., *CheckAll ()* in Table 1) that calls the *QueryPrice* web services implemented by all qualified suppliers performs the following tasks:

Table 1 Partial program listing of the price checking function

1	Private Sub CheckAll()
2	Dim ProductSupplierRow As DsProductSupplier.productsupplierRow
3	Dim myQueryPrice As QueryPrice.Service1
4	Dim QueryResult As New QueryPrice.ProductQuery()
5	Dim con As OleDb.OleDbConnection
6	Dim cmd As OleDb.OleDbCommand
7	Dim wsdl As String
8	con = New OleDb.OleDbConnection(“PROVIDER=Microsoft.Jet.OLEDB.4.0;” _
9	& “Data Source=” & MapPath(“Inventory.mdb”)
10	cmd = New OleDb.OleDbCommand()
11	con.Open()
12	cmd.Connection = con
13	For Each ProductSupplierRow In DsProductSupplier1.productsupplier.Rows

Table 1 Partial program listing of the price checking function (continued)

```

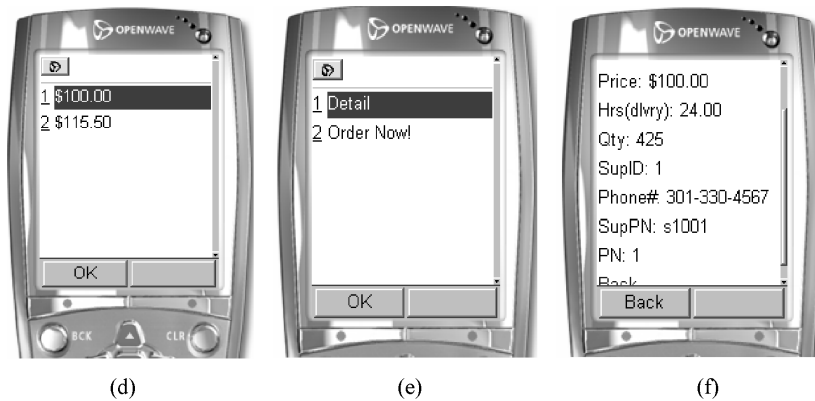
14 Try
15     cmd.CommandText = "select wsdl from suppliers where " & _
16     " wsoperation = 'Check' and supid = " & ProductSupplier.supid
17     wsdl = cmd.ExecuteScalar()
18     myQueryPrice = New QueryPrice.Service1(wsdl)
19     QueryResult = myQueryPrice.Check(ProductSupplier.supPN)
20 Catch ex As Exception ' Error in query the price
21     QueryResult.price = -1
22     QueryResult.QuantityOnHand = -1
23     QueryResult.DeliveryTime = -1
24 End Try
25 ProductSupplierRow.price = QueryResult.price
26 ProductSupplierRow.quantity = QueryResult.QuantityOnHand
27 ProductSupplierRow.deliveryTime = QueryResult.DeliveryTime
28 Next
29 End Sub
30 'Data structure matches the SOAP response of the Check web method
31 Public Class ProductQuery
32     Public price As Double
33     Public QuantityOnHand As Integer
34     Public DeliveryTime As Single
35 End Class

```

- An instance, *myQueryPrice*, of that web service is declared (Line 3 in Table 1).
- A private directory is used to store the URLs of WSDL files for web services published by suppliers. The WSDL files defining the *QueryPrice* web services are retrieved from the directory (Line 15–17).
- The retrieved WSDL files are used to instantiate the *QueryPrice* web service of each qualified supplier via dynamic binding (Line 18). The *Check* method of this web service is then invoked using product ID as a parameter (Line 19). Product IDs of each supplier are stored in an internal Supplier-Product database table.
- The result of a price inquiry contains the product price, quantity of the product on hand, and an estimated delivery time (in hours). It is stored as a record (i.e., *ProductSupplierRow*) in the *ProductSupplier* table (Line 25–27) in an in-memory *DataSet* called *DSProductSupplier1*. The *ProductSupplier* table is bound to *ObjectList*, a data-aware mobile form control for displaying data. The rendering of pricing information from all suppliers is handled by an instance of *ObjectList* as illustrated in Figure 5(d)–5(f). Figure 5(d) only displays the offered prices from two different suppliers in the ascending order.

After price quotes from all qualified suppliers are returned, the user can select a price. Then a page with a link to detailed information and a link to the ProductOrdering service will be displayed (see Figure 5(e)). The user can press the BCK button to go back to the previous page. If the *Detail* link is selected from the Figure 5(e), the detailed product-offering page will be displayed (see Figure 5(f)).

Figure 5 (d) Displaying prices from vendors, (e) Viewing a detailed offer or placing an order and (f) Detailed offer record from a supplier



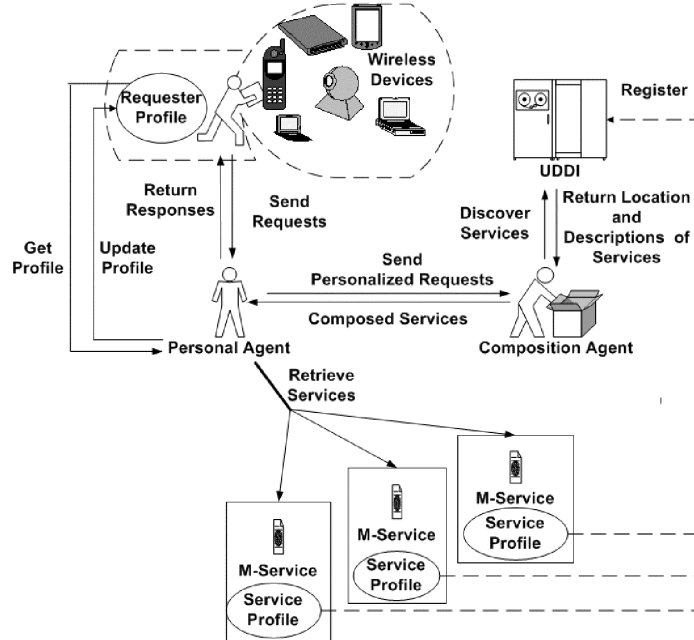
After checking all the details including price, delivery commitment, and availability provided by suppliers, the purchasing personnel can either call a selected supplier to place an order via the cell phone, or invoke an ordering method of the ProductOrdering web service provided by the supplier to place an order online.

The e-procurement m-service uses two standardised web services for dynamic price checking and ordering at runtime. If companies want to join the e-procurement system as suppliers, they need to implement these two web services using the same method names and signatures in accordance with the WSDL files. The e-procurement system administrator only needs to add new entries for WSDL files to the private directory in order to make their services available. No change to the application source code is required.

5.3 Intelligent m-services

A combination of web services and intelligent agents is expected to re-shape the future service industries by encouraging the federation of workforces among organisations. Distributed heterogeneous intelligent agents can interact with each other in various ways such as peer-to-peer and agent-mediated manners [20].

To further enhance the Agency component in the m-service portal, we propose an architecture for intelligent m-service agency as depicted in Figure 6. It includes a variety of coordinated agents, such as personal agents and composition agents that interact with other major components in the m-service portal such as the Profiler, as well as external m-service registries and m-services.

Figure 6 An intelligent m-service agency architecture

- *Personal agents* facilitate service discovery and use. When a user requests for a new service, he only needs to authenticate his credentials with his personal agent. The personal agent can automatically acquire his personal information from the Profiler and then provide it to appropriate service providers. In addition, the personal agent can dynamically update user's profile based on his recent requests. If the personal agent finds that the user is requesting a previously consumed service, the agent can automatically retrieves his prior access information and preferences/constraints for that service. If the requested service is new, the personal agent will update the corresponding profiles.
- *Composition agents* search and aggregate multiple m-services before returning them to a personal agent. Several advantages can be gained from the use of composition agents [21]. In the intelligent m-service agency architecture, composition agents provide the capability of creating new services by reusing existing ones. They also allow personal agents to focus on maintaining requesters' profiles and retrieving services.

It is well recognised that intelligent agents are characterised with automation and pro-activation. As long as a user specifies his/her service interest in a requester profile, the agents can take the initiative to search for information about relevant services. Agents can notify the user whenever the requested service is available. Moreover, they can support the mobility and implementation of service models. The mobility offered by an agent-supported flexible infrastructure is capable of identifying and tracing mobile entities and adapting them to the changing network condition. However, mobility of agents in open environments requires an adequate level of security and the ability of handling system heterogeneity [22].

Achieving the goal of combining agent technology and m-services presents many challenges, including semantic interoperability, agent-based collaboration, service customization, as well as automated and flexible service discovery and transaction across the network. One crucial challenge is to make services understandable by heterogeneous agents. Another challenge is to enable mapping between different annotations of web services. Both challenges can be boiled down to the issue of precise interpretation of web services, which is a necessary prerequisite for automatic web service discovery and composition. By using common semantics, a service annotation language can offer a high degree of flexibility and expressiveness. Adding semantics to service annotations can reduce ambiguity in interpreting service information and improve both quality and quantity of e-commerce transactions [19]. Driven by the demand of more effective web service discovery and composition, the necessity of developing standardised ontology and service profiles has been well recognized. The semantic web [23] provides an infrastructure to support a service description language that improves query matching, negotiation proposals and agreement, and advertisement [24]. Therefore, a mechanism needs to be established to:

- enhance the description and composition of web services (e.g., DAML-S [25]) and the representation of users' profiles and preferences
- support the process of searching for m-services [19].

Our e-procurement prototype can potentially be enhanced by incorporation of the semantic web. The composition of web services is better to be accomplished dynamically with declarative means (e.g., [26]) rather than with procedural code. Domain-specific ontologies developed for individual industries and trading groups can support service interoperability and request-service matching. A profiler may benefit most from semantic web services in view of the number and variation of users' profiles. A standard representation of profilers may help not only discover web services, but also create user communities. Ultimately, users that share common profilers will be able to filter services collaboratively and contribute to profiler services collectively.

6 Discussion

M-services are still in a very early stage. They require dynamic web service interoperability in real-time to match web service availability, mobile device capability, preferences of individual users, wireless network bandwidth, and so on. Among many issues that remain to be fully investigated, a few of them are listed as follows:

- *Platform interoperability.* Since web services are normally heterogeneous entities, how to effectively connect and coordinate them to build integrated services is a critical issue [27].
- *Context-aware m-services.* Traditional wireless applications lack the context data required to handle even relatively simple customisations, depriving wireless applications of the ability to handle desired social nuance and control of individuals. Context can be defined as any information that characterises a situation related to the interaction between humans, applications and the surrounding environment [28]. Typical examples of contextual features are location, time, speaker, hearer, other

salient agents, and objects. Context-aware m-service delivery aim to offer the promise of significant user gains – an ability of a service to adapt to a changing environment automatically.

- *Mobile client caching.* Client caching refers to the ability of mobile clients to retain copies of data items locally once they have been obtained from the server. Caching of frequently accessed data will play an important role in m-services because of its ability to alleviate the problems of long transmission latency and unreliable networks during weak-connections and disconnections [29].
- *Security.* End-to-end security between the internet and mobile devices is critical in m-commerce. Security channels between the client and the WAP gateway are supported by WTLS (Wireless Transport Layer Security), while the channel between the WAP gateway and web server is supported by TLS (Transport Layer Security). Translation between WTLS and TLS is executed at the WAP gateway. The WTLS specification does not provide this level of security. Therefore, how to develop a reliable solution for authenticating users across a wide variety of mobile devices and applications is critical.

In this paper, we have proposed an m-service portal architecture and exemplified it with an e-procurement m-service prototype. An intelligent m-service approach is introduced to improve the adaptability and flexibility of the m-service portal. We also discussed several issues that need to be further explored, such as the development of domain specific ontology, the implementation of intelligent m-services, and effective composition of web services. In most cases, users of mobile devices gain access to web services via a web service client (such as an HTTP-SOAP client). Web Services for Remote Portals (WSRP) is an emerging standard for building user-facing web services for portals [30]. Business Process Execution Language for Web Services (BPEL4WS) provides a standard way of specifying web service composition [31]. We will be exploring the integration of semantic web research with emerging web service standards such WSRP and BPEL4WS in the context of m-service portals. The preliminary evaluation has indicated that the proposed m-service portal provides adaptability, flexibility, and easy access to m-services for mobile workforces.

References

- 1 Wrolstad, J. (2002) *Mobile Workforce Nears 100-Million Mark*, <http://www.wirelessnewsfactor.com/perl/story/18453.html>.
- 2 Intel (2002) *Wireless Technology for Enterprises: High Payback Opportunities*, August, http://www.intel.com/pca/developernetwork/doc/horiz_clarity.pdf.
- 3 Maamar, Z., Benatallah, B. and Sheng, Q. (2002) 'Towards a composition framework for e/m-services', *Proceedings of the UbiAgents Workshop'02*, Bologna, Italy.
- 4 Vecchio, D. (2001) 'Legacy software: junkyard wars for web services?', *Proceedings of the Gartner Symposium IT Expo Presentation*, Orlando, Florida.
- 5 Varshney, U. and Vetter, R. (2002) 'Mobile commerce: framework, applications and networking', *Mobile Networks and Applications*, Vol. 7, pp.185–198.

- 6 Dave, B. and Patterson, L. (2002) 'Extending enterprise applications to mobile users', July, http://www-3.ibm.com/software/pervasive/tech/pdf/PVC_WP.pdf.
- 7 Intel (2002). *IT Mobility Road Map: Past, Present, and Future Plans – Intel Information Technology Worldwide*, April, <http://www.intel.com/eBusiness/pdf/it/pp021802.pdf>.
- 8 Zetie, C. (2002) *Essential Elements of a Mobile Enterprise Strategy*, Giga Information Group.
- 9 Siau, K., Shen, Z. and Varshney, U. (2003) 'Communications and mobile services', *International Journal of Mobile Communications*, Vol. 1, Nos. 1–2, pp.3–14.
- 10 Yuan, Y. and Zhang, J.J. (2003) 'Towards an appropriate business model for m-commerce', *International Journal of Mobile Communications*, Vol. 1, Nos. 1–2, pp. 35–56.
- 11 Louis, P.J. (2001) *M-Commerce Crash Course*, McGraw-Hill Companies.
- 12 Sharma, S.K., Kitchens, F.L. and Miller, P. (2003) 'A new approach for mobile communication: web services using flexible services architecture', *International Journal of Mobile Communications*, Vol. 1, No. 3, pp.301–311.
- 13 Wohl, A.D. (2002) *Is There a Killer App for Mobile?*, <http://www.wohl.com/wa0231.htm>.
- 14 Hanebeck, H-C.L. and Tracey, B. (2003) 'The role of location in supply chain management: how mobile communication enables supply chain best practice and allows companies to move to the next level', *International Journal of Mobile Communications*, Vol. 1, Nos. 1–2, pp.148–166.
- 15 Varshney, U. (2003) 'Location management for wireless networks: issues and directions', *International Journal of Mobile Communications*, Vol. 1, Nos. 1–2, pp.91–118.
- 16 Varshney, U. (2003) 'Issues, requirements and support for location-intensive mobile commerce applications', *International Journal of Mobile Communications*, Vol. 1, No. 3, pp.247–263.
- 17 Roussos, G., Tuominen, J., Koukara, L., Seppala, O., Kourouthanasis, P., Giaglis, G. and Frissaer, J. (2002) 'A case study in pervasive retail', *Proceedings of the Second International Workshop on Mobile Commerce*, Atlanta, Georgia, USA, pp.90–94.
- 18 Zhang, D. (2003) 'Delivery of personalised and adaptive content to mobile devices: a framework and enabling technology', *Communications of AIS*, Vol. 12, August, pp.183–202.
- 19 McIlraith, A.S., Son, T.C. and Zeng, H. (2001) 'Semantic web services', *IEEE Intelligent Systems, Special Issue on the Semantic Web*, Vol. 16, No. 2, pp.46–53.
- 20 Sycara, K. (2002) 'Agents supporting humans and organisations in open, dynamic environments', *Proceedings of the AAMAS'02*, Bologna, Italy, p.1043.
- 21 Maamar, Z., Mansoor, W. and Mahmoud, Q.H. (2002) 'Software agents to support mobile services', *Proceedings of the AAMAS'02*, Bologna, Italy, pp.666, 667.
- 22 Bellavista, P., Corradi, A. and Stefanelli, C. (2000) 'A mobile agent infrastructure for the mobility support', *Proceedings of the ACM Symposium on Applied Computing (SAC'00)*, Como, Italy, pp.539–545.
- 23 Berners-Lee, T., Hendler, J. and Lassila, O. (2001) 'The semantic web', *Scientific American*, pp.35–43.
- 24 Trastour, D., Bartolini, C. and Preist, C. (2002) 'Semantic web support for the business-to-business e-commerce lifecycle', *Proceedings of the WWW 2002*, Honolulu, Hawaii, USA.
- 25 Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D.L., McIlraith, A.S., Narayanan, S., Paolucci, M., Payne, T., Sycara, K. and Zeng, H. (2001) 'DAML-S: semantic markup for web services', *Proceedings of the International Semantic Web Working Symposium (SWWS)*.
- 26 Narayanan, S. and Mallraith, S.A. (2002) 'Simulation, verification and automated composition of web services', *Proceedings of the WWW'02*, Honolulu, Hawaii, USA, pp.77–88.

- 27 Benatallah, B., Dumas, M., Sheng, Q.Z. and Ngu, A.H.H. (2002) 'Declarative composition and peer-to-peer provisioning of dynamic web services', *Proceedings of the 18th International Conference on Data Engineering (ICDE'02)*, San Jose, CA, USA.
- 28 Dey, A.K. and Abowd, G.D. (2001) 'A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications', *Human-Computer Interaction*, Vol. 16, No. 24, pp.97–166.
- 29 Kahol, A., Khurana, S., Gupta, S.K. and K. Srimani, P. (2000) 'An efficient cache maintenance scheme for mobile environment', *Proceedings of the International Conference on Distributed Computing Systems*, pp.530–537.
- 30 OASIS (2003) *OASIS Web Services for Remote Portals (WSRP) TC*, <http://www.oasis-open.org/committees/wsrp/>.
- 31 Thatte, S. (Ed.) (2002) *Business Process Execution Language for Web Services*, 1.0 ed., <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>.