

## WEB SERVICES APPLIED TO MOBILE ENVIRONMENTS

Alexandru PETCULESCU<sup>1</sup>

*În acest studiu propunem un framework de servicii web dedicat mediilor mobile și analizăm beneficiile asupra lucrului colaborativ în medii mobile. Scopul acestei lucrări este de a defini un framework dedicat implementării serviciilor mobile pentru echipamente de tipul smartphone. Mediul de comunicație folosit este standardul XML și SOAP pentru a putea avea interoperabilitate.*

*This paper presents a new concept of a web services framework for mobile environments where the service providers are abstract to the invoking client thus allowing the environment to be dynamic – service providers may come and go transparent to the client. The purpose of this paper is to define architecture of the framework used for implementing mobile services for handheld devices like smartphones. Communication platform used is XML/SOAP standard for interoperability.*

**Keywords:** web services, smartphones, mobile, adhoc network

### 1. Introduction

During past years an increase in mobile technology use occurred together with the increase in performance and functionalities for handheld devices like smartphones, PDAs or the newly comers iPads. Meanwhile Software as a Service concept offers more and more support for mobile devices which now use software previously adapted for PCs only and these devices need a reliable framework to allow them to consume the services offered.

Mobile era 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. Mobile 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.

Mobile commerce offers many business opportunities. Wireless content delivery service is a critical element of mobile commerce. Popular content-

---

<sup>1</sup> PhD Student, Computer Science Department, University POLITEHNICA of Bucharest, Romania, e-mail: alexp@microsoft.com

oriented mobile commerce applications are financial news delivery, stock portfolio tracking, stock quote checking, and telephone directory assistance. Marketing applications in mobile commerce may automatically alert users of special sales at nearby shops based on their locations and interests. Transaction-oriented mobile commerce such as trading stocks often requires immediate action even when people are traveling. Mobile 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 mobile services. Mobile 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 mobile services may reduce data entry operations for the mobile user in areas such as supply chain management. Using location-based mobile services to advertise sales information to customers at the proximity of stores may attract more customers and increase the revenue. Innovations in B2C mobile applications can be found in some vertical industries such as retailing. 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.

Today issues regarding smartphones interoperability, discovery of services based on semantics, acquiring and distributing services after their discovery are not extensively researched. However these problems may be overcome using web services that allow loosely coupled connections.

The goal is to have an active and dynamic environment where smartphones can freely access the services provided over the network by their owners. This environment should fulfill the needs of mobile communication where service providers go online and offline randomly but the services should be available at any time transparent to changes.

In the following chapters we present the basic idea of this web services framework and we show how web services are enclosed into an abstract entity so a client request will be handled by an available physical representation of the abstract service.

## **2. Smartphones and web services model**

In post PC era there is a trend to miniaturize computing devices with greater computing power and network connection capabilities.

### **Smartphones**

Smartphones characteristics that make them ideal for collaborative working environment are the following:

1. Network connectivity: most smartphones have more communication interfaces like Bluetooth, GPRS and Wi-Fi.
2. Mobility: because of wireless networks and smartphones people are moving from a tethered computing environment towards mobility allowing users to work even on the move.
3. Multimedia: smartphones are capable of recording and play backing both audio and video.
4. Virtualize: applications available on smartphones are similar in functionality to desktop applications to allow people to work anytime anyplace. E.g.: Microsoft Office for Windows Mobile.

### **Web Services model**

In our days most of the companies offer their services over the Internet. Web services are applications that adopt a universal language to send data and instructions over Internet. Their goal is to provide a cost effective solution for companies who conduct their business on the Internet. Programmers can use their favorite programming language or operating system to describe and implement Web services that may be invoked over the web later on.

### **Classic and distributed model: differences**

Comparing interactions in web services based systems we have three components: a service provider, a service requestor and a service catalogue. Sometimes another participant – the service broker – may exist [1]. Interactions in this mobile environment are different than interaction in traditional wired network environment in the following aspects:

Dynamic environment – services in mobile networks are dynamic, providers come and go and users need to be updated with the changes in the network.

Limited resources – smartphones have limited autonomy in terms of power, computing and storage as well as a reduced display. A service requestor running in a mobile environment must use less power to fulfill a request than a traditional consumer.

Heterogenic environment – a mobile device may be a service source, a consumer source or both [2].

High degree in dynamicity and heterogeneity of mobile environments needs redesigning or extension of traditional service discovery solutions, invocation and management to allow a more intelligent search for services as well as providing personalized services based on user's context. In order to allow service oriented applications to reach mobile environments a service framework is required to manage, deliver and discover efficiently the provided services.

The purpose of this paper is to propose architecture of a framework used by the services requested by mobile devices in order to keep these services more adaptable and accessible. In this paper a mobile services framework that delivers a series of functions like service registration, discovery and invocation as well as services management is presented.

### **3. Related work**

Universal Description Discovery and Integration offers fundamental support for publishing, discovering and management of web services in a distributed environment. The core of UDDI is a service registry with the following components:

- White pages – contact of business entities and service identifiers
- Golden pages – classification information of business and service
- Green pages – technical information about the services.

UDDI offers developer interfaces to query and update the information related to the service. However service metadata discovery is an operation that takes place at development time. The support for dynamic services is limited. More than that, issues like fault tolerance and load balancing need deeper research.

In order to provide adaptive and personalized services to match the constraints of mobile services environments, Minder Chen proposed a mobile services portal. This portal interacts with mobile services registries and supports mobile devices invocations of the web services. The portal manages preconfigured mobile web services as well as personal data reducing the interaction between services and mobile agents. The portal also support automatic discovery of services. A limitation of this portal is the ability of working with homogenous systems. Another limitation is to efficiently find and connect available services [3].

In order to efficiently support semantic based services discovery Sonia Ben Mokhtar proposed an EASY solution which implies EASY-Language and EASY-Matching. EASY-L is a language for semantically services description. EASY-M defines a set of relations and an order to evaluate them to find the required service. EASY reduces a lot the number of matches necessary for a service request. The EASY purpose is to offer an efficient mechanism for service discovery. EASY-L is an extensible language specified in OWL making service properties describing complex [4].

DAFNE is a distributed framework for service management for ambient reconfigurable services in mobile paradigm. DAFNE allow users to register their own services covering dynamic development, deployment and discovery of the services together with service orchestration in order to have easy service

interoperability [5]. On the other hand the support for service changes detection is limited [6].

AIDAS is a middleware that supports mobile services' both discovery and configuration management. Service discovery and selection based on user's context information are supported [2]. AIDAS uses semantics-based metadata to describe the properties and characteristics of the services and clients involved in discovery. A downturn of this middleware is the large volume of storage resources used.

#### **4. Mobile services framework**

##### **Basic idea**

The basic idea behind this concept of a mobile services framework is to provide mobile environment a platform that allows service consumers to use network services independently of network changes. This means that a service provided to clients should be available even if one of the service providers goes offline. In order to achieve that, we introduce a middle layer where a physical service is enveloped in an abstract service together with several other providers for the same service type.

We propose a dynamic service framework to ease service-oriented application development in mobile environments. The following functionalities will be supported by the framework:

- Heterogeneous components integration – inside the framework the components involved in mobile service oriented application are packed in web services. The framework offers an infrastructure to integrate and coordinate services so that the consumer of the service as well as its provider can interact smoothly in a heterogeneous environment using messages or events mechanism.
- Dynamic service invocation – Mobile services aren't stable because the dynamic characteristics of mobile environments. The platform dynamically selects available services accordingly to service consumer's requirements, service states and properties. Additionally to reduce the consumption of the limited resources the framework dynamically balances payload among similar web services.

##### **Architecture**

The architecture of the service framework includes the following components: administration console, service broker, event dispatcher, metadata repository, semantic data repository, discovery service, invocation service, security controller.

Administration console provides a user friendly interface to query, register and monitor services. Service broker collaborates with discovery service and invocation service to dynamically query the services and to balance the service payload. Metadata repository contains services' static and runtime data. Semantic repository contains semantic tags used for service registration and discovery. Security controller is a tool for authentication and authorization of the users. Event dispatcher tracks service changes and notifies the administration console and service broker about changes in service state.

The platform supports service oriented applications in mobile environments and covers the aspects of resource registration, service discovery, invocation, monitoring and administration. It dynamically collects data and executes corresponding actions.

Innovative features of the framework:

- Uniform interface – the framework treats any entity as a resource. It isolates service consumer from the changes on the service provider side. The metadata repository contains the id, Uri, WSDL and the properties related to service state. Based on service discovery information it dynamically selects a service sent to the invocation service which dynamically accesses the service based on service states, service properties or user's context.
- Abstract resources– the framework abstracts the resources and allows service consumers to access the services through uniform service interfaces. Initially a resource is a conceptual mapping of a set of descriptors that finally maps to a physical representation of the resource. Behind a resource is a set of similar candidates and one of them will qualify to answer the request. The abstract resource is kept in metadata repository. When a consumer initiates a request, the request is delivered to the service broker. The service broker locates its abstract representation in metadata repository. The abstract resource contains the abstract service and its corresponding physical representations. Having this correspondences schema the broker calls the discovery service to find the available and invokes the qualified service. Dynamic invocation service balances the requests to reduce resource consumption.
- State changes events – mobile environments are dynamic environments where entity state may change from a second to another. If these changes are caught by the service broker at the right time then service unsuccessful calls may be avoided. In this framework the event dispatcher receives state changes events and triggers the necessary events. There are four types of events:

- Service\_Added\_Event: fired when a service or abstract service is added in the framework.
- Unavailable\_Service\_Event: triggered when a service becomes unavailable.
- Service\_End\_Event: triggered when a service or abstract service is removed from the framework.
- Changed\_Property\_Event: fired when any of the properties of a service or abstract service changes.

The event dispatcher periodically sends „ping” signals to the registered services to query the state of the services. As services state changes the dispatcher event facilitates the broker job to select qualifying services.

Below is a diagram that presents the elements implied in this framework:

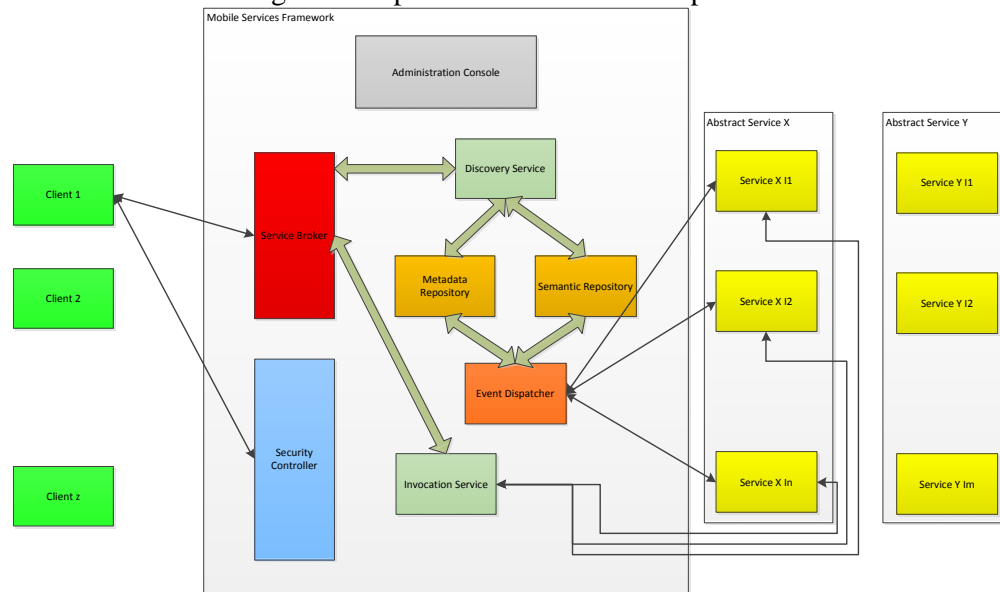


Fig. 1. Platform Architecture

## Scenario

The scenario for using the mobile service framework is the following:

### Service Registration

1. Service X I1 physical web service registers to the framework sending a Service\_Added\_Event, caught by “Event Dispatcher”. The metadata contained by the register packet describes abstract “Service X”.

2. The Event Dispatcher extracts the identification metadata and sends it to “Metadata Repository”. The semantic metadata is sent to “Semantic Repository”.

3. Other instances of the same abstract service X may be added the same way as Service X II.

4. Other abstract services may be added the same way as abstract “Service X”.

### **Client Registration**

1. Client 1 registers to the framework interacting with the “Security Controller”.

2. The same way other clients may register to the framework

### **Service Invocation**

1. Client 1 needs to issue a request for abstract Service X.

2. Client 1 sends a request to the “Service Broker”.

3. The “Service broker” queries the “Discovery service” for the physical service representation that would treat the request.

4. The “Discovery service” queries the “Semantic repository” and “Metadata repository” and selects the service that response the request best and returns it’s details to “Service Broker”

5. The service broker sends the details to the “Invocation service” which calls the physical service.

6. The result is sent back to “Service broker” and then to the client.

### **Updating services repository**

The data about available abstract and physical services is updated by the “Event dispatcher”.

1. When Service\_Added\_Event, Unavailable\_Service\_Event, Service\_End\_Event, Changed\_Property\_Event is fired “Event dispatcher” catches the event, extracts required information and updates data in “Metadata repository” and “Semantic repository”

2. Periodically the “Event dispatcher” iterates the services subscribed in “Metadata repository” to check their availability. If a service is discovered to be unavailable the metadata is updated in “Metadata repository”.



### Performance evaluation

The performance degradation problems in a mobile computing environment face the same issues as those in conventional distributed computing environment. One of the first challenges is the attempt to provide an efficient message scheme while using XML.

Following we have the overhead of ASP and web services implemented using Microsoft technologies:

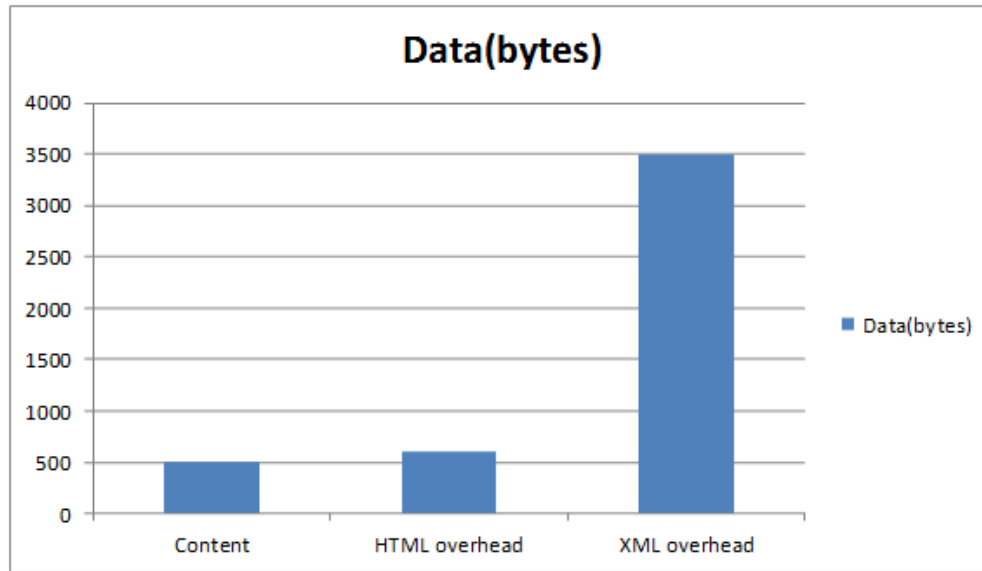


Fig. 2. Comparison of data sizes

For improving the XML overhead in our case we decided to compress XML because the CPU overhead on today's handheld devices is less than the network latency.

In terms of framework evaluation we compared a traditional web service implementation with our own, both of them being consumed by a handheld device. We implement to benchmark using two different data types as parameters to the web service: float data type and string data type. We measure a total session time and a message size for the service call.

- String array concatenation – is a service that creates a single string from all the strings in the array.
- Floating number addition – is a service that receives an array of floating numbers and returns the sum of all the numbers in the array.

The testing environment used for the comparison of traditional web service vs. our idea was a virtual lab containing a Microsoft Windows 2008 server with

IIS 7.0 and .NET Framework 3.5. The web services were implemented using ASP.NET technology and the development language used was C#.

Below are the preliminary results obtained by running tests in the environment described above. Table 1 shows the messages size in bytes for string concatenation and float addition services:

Table 1

Number of elements in the array	Message size(bytes)				Time(Milliseconds)			
	Traditional Web Services		Mobile Service Framework		Traditional Web Services		Mobile Service Framework	
	String concatenation	Floating number addition	String concatenation	Floating number addition	String concatenation	Floating number addition	String concatenation	Floating number addition
2	350	360	402	403	780	781	845	803
3	371	385	426	435	828	781	866	823
4	395	405	443	458	828	933	890	942
5	418	435	466	488	922	1027	934	1010
6	443	461	501	507	890	956	920	989
7	465	487	527	532	860	873	922	912
8	493	515	579	601	984	878	978	902

As we can see in the above table there is a slight difference between those two implementations. The overhead in time comes from the discovery part of the service which is independent of the data transferred so it may be considered as a constant  $C_t$ . The overhead in size comes from the user dependent context data and is also independent of the necessary data for the operation or the operation so it may also be considered as a constant  $C_s$ .

The  $C_s$  constant may be reduced using XML compression for the messages transferred between client and server and between server and services provider.

## 5. Case study

In a week-end day there are about 50.000 customers in a shopping center in Bucharest. These clients look for shopping, dining or entertaining. For these customers the shopping center provides location based querying services.

When a client sends a request to the service – E.g. A 23 years old woman queries the service for fashion stores located at 3rd floor – the service broker queries a service based on the abstract service and the information is returned as in below figure – in our case the information returned is about fashion stores for young women:

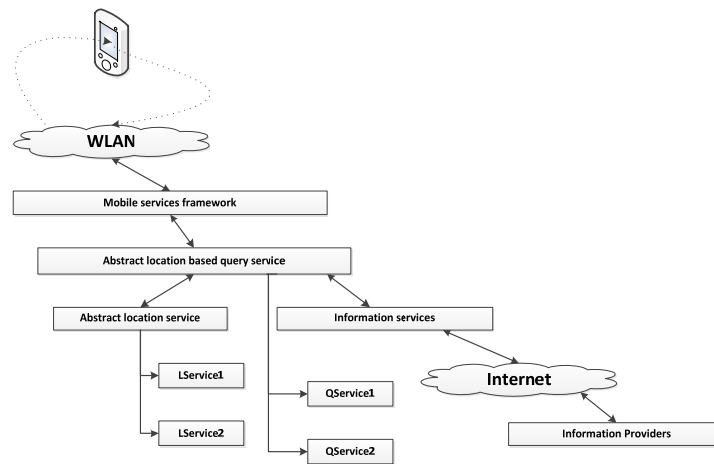


Fig. 3. Case study

This case study is purely theoretical based on demographical statistics.

## 6. Conclusion

As the mobile environments become more and more important in our daily life, it becomes more and more imperative the need for a framework that offers continuity and stability in services provided.

In the paper we presented a mobile services framework for web services in mobile environments. The framework supports uniform service interface, abstract resource service invocation and service state changes and facilitates services access from handheld devices.

In future problems like service selection based on semantics and low cost semantic matching for service selection will be addressed.

## REFERENCES

- [1] J. O'Sullivan, D. Edmond and A.T. Hofstede, "What's in a service? Towards Accurate Description of Non-functional Service Properties", *Distributed and Parallel Databases*, Vol. 187, No. 3, 2008
- [2] A. Toninelli, A. Corradi and R. Montanari, "Semantic-based discovery to support mobile context-aware service access", *Computer Communications*, **Vol. 31**, No. 5, 2008
- [3] M. Chen, D.S. Zhang and L.N. Zhou, "Providing web services to mobile users: the architecture design of an m-service portal", *International Journal of Mobile Communications*, **Vol.3**, No.1, 2005
- [4] S.B. Mokhtar, D. Preuveneers, N. Georgantas, V. Issarny and Y. Berbers, "EASY: efficient semantic service discovery in pervasive computing environments with QoS and context support", *Journal of Systems and Software*, **Vol.81**, No.5, 2008

- [5] *V. Stavroulaki, K.P. Demestichas, E.F. Adamopoulou and P. Demestichas*, “Distributed web-based management framework for ambient reconfigurable services in the intelligent environment”, *Mobile Networks and Applications*, **Vol. 11**, No.6, 2006
- [6] *Z. Maamar, H. Yahyaoui and Q.H. Mahmoud*, “Dynamic management of UDDI registries in a wireless environment of web services: Concepts, architecture, operation, and deployment”, *Journal of Intelligent Information Systems*, **Vol.18**, No.2, 2007