Mobile Web Services for Mobile Commerce

Subhankar Dhar
San Jose State University, USA

INTRODUCTION

In recent years, Web services have gained popularity in terms of applications and usage. A copious volume of literature has been published that describes the potential benefits of this technology (Chu, You, & Teng, 2004; Gehlen & Pham 2005; Pasthan, 2005; Yoshikawa, Ohta, Nakagawa, & Kurakake, 2003). The business community has slowly started to realize how this technology can be used to solve various business problems (Tilley, Gerdes, Hamilton, Huang, Muller, Smith, & Wong, 2004). The idea of a Web service stems from the fact that different applications developed in heterogeneous technology platforms can be seamlessly integrated together via some common standard protocols over World Wide Web (Chu, et al., 2004; Pasthan, 2005). This will facilitate reusable software component development, enterprise application integration (EAI), and distributed application development.

A new emerging area that is currently under development is mobile Web services, which will be quite useful in terms of capability and applications. There are numerous business benefits in using these mobile services, but these new technological developments pose various challenges. In this article, we discuss emerging mobile commerce applications and then present an open mobile Web services framework that will support various mobile commerce applications.

WEB SERVICES AND SERVICE-ORIENTED ARCHITECTURE

What are Web Services?

The W3C defines Web services as follows: "A Web service is a software system identified by a URL, whose public interfaces and bindings are described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols" (W3C, 2006).

At a conceptual level, Web services consist of a service provider, a service requestor, and a service registry, as shown in Figure 1. The service requestors and providers communicate with each other by exchanging messages using open standards and protocols. Another important feature is that the requestors and providers form loosely coupled systems, meaning that the development of each of the systems is done in a truly distributed manner. They are developed, implemented, and maintained independent of each other.

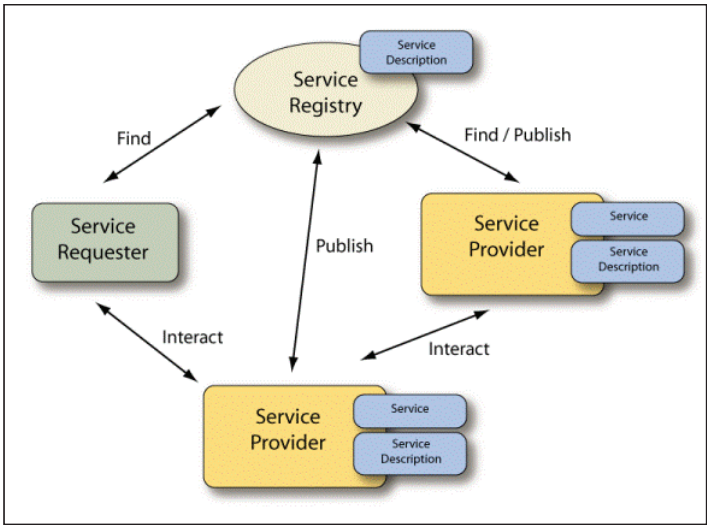
Web Services Standards and Protocols

Web services is a layered stack of protocols that consists of the service transport layer, the XML messaging layer, the service description layer, and the service discovery layer. This is illustrated in Table 1.

The function of the XML messaging layer is to encode messages in a common XML format so that they can be interpreted by all communicating services, which is independent of hardware platform or implementation language. The simple object access protocol (SOAP) (Gao et al., 2004; Mnaouer, Shekhar, & Liang, 2004) can interpret regular XML and invoke a Web service. In addition, it can also retrieve the results in a consistent manner. The entire SOAP message is wrapped in an XML envelope and is sent using HTTP.

The function of the service description layer is to describe the public interface to a specific Web service. This is done using the Web service description language (WSDL), which is essentially an XML grammar. The public interface to a Web service typically includes all methods provided by the service along with their invocation requirements. In addition, the public interface contains binding information to specific transport protocols, and address information for locating the service. A client can locate a Web service and invoke

Figure 1. Web services model



any of its available functions using WSDL. There are various WSDL-aware tools available that will automate this process, thereby enabling an application to easily integrate new services with very little coding.

The function of the service discovery layer is to register Web services into a common registry. The universal description, discovery, and integration (UDDI) protocol is responsible for service discovery. Data in UDDI are stored in XML format. The users and various other Web services will be able to locate Web services through UDDI. After a Web service is found, the Web service description in WSDL format is retrieved and then the service is invoked by SOAP.

Service-Oriented Architecture

A service-oriented architecture (SOA) is a collection of service providers that has public interfaces through which the services are assessable by a third party (Erl, 2004). A service is created by means of an application that has an interface through which messages (typically XML) can be transferred. A service also encapsulates its data, and manages ACID (Atomic, Consistent, Isolated, Durable) transactions within its data sources. The individual service characteristics make SOA highly suitable for Web services.

Services are typically developed in the following four layers:

- Client interface
- Business process interface
- Business rules
- Data access

The client interface is the bridge between clients of a service and the service's business process interface. A single service can have multiple interfaces, such as Web services, queuing systems, or simple file shares. The

Table 1. Web services protocol stack

SERVICE	PROTOCOL
Discovery	UDDI
Description	WSDL
XML Messaging	XML-RPC, SOAP
Transport	HTTP, BEEP
Network	TCP, IP