Mobile Shopping Assistant: Integration of Mobile Applications and Web Services

Huaigu Wu
Huaigu.Wu@sap.com

Yuri Natchetoi
Yuri.Natchetoi@sap.com

SAP Labs
Montréal, Québec, Canada

ABSTRACT

The goal of this poster is to describe our implementation of a new architecture enabling efficient integration between mobile phone applications and Web Services. Using this architecture, we have implemented a mobile shopping assistant described further. In order to build this architecture, we designed an innovative XML compression mechanism to facilitate data exchange between mobile phones and Web Services. We also designed a smart connection manager to control asynchronous communication for all possible channels of a mobile phone. In addition, we used diverse input modes in order to extend users' access to Web Services.

Categories and Subject Descriptors

C.1.4 [**Parallel Architectures**]: [Mobile Processors]; H.5.3 [**Group and Organization Interfaces**]: [Collaborative computing, Asynchronous interaction]; E.4 [**CODING AND INFORMATION THEORY**]: [Data compaction and compression]

General Terms

Design, Management

Keywords

Collaborative system, Mobile application, XML Compression, Asynchronous Communication

1. INTRODUCTION

Undoubtedly business applications running on mobile devices will use more and more Web Services because integration of mobile applications and Web Services is a win-win strategy. Since mobile devices cannot process complex business logic for mobile applications due to their limited capabilities, Web Services can provide the necessary processing power. In turn mobile applications will provide a huge consumer market for Web Services.

Nowadays cell phones are the most popular mobile devices. However, until now, it is still very difficult to find mobile phone applications that access Web Services. There are some obstacles: 1) unstable network connection status limiting the mobile phones ability to invoke Web Services efficiently; 2) limited bandwidth of mobile communication channels that limits the ability to transmit SOAP messages; and 3) lack of an efficient XML parser used to process SOAP messages on the client. Although there is currently existing research on each of these problems, we are not aware of any integrated architecture that solves all of them. In this poster, we propose an integration architecture that provides an efficient Web Services integration with mobile applications, addressing all the above issues.

In order to prove the concept, we built a mobile shopping assistant based on the proposed architecture. Our implementation solves the aforementioned problems by using an innovative XML compression algorithm and a smart connection manager for communication channels. We also show how to use diverse input modes of mobile phones, such as camera and voice.

2. MOBILE SHOPPING ASSISTANT

The mobile shopping assistant is a J2ME application running on regular mobile phones. When a customer is walking into a store, she[1] can use the shopping assistant to access Web Services published by the store. Some examples of Web Services that a store can publish are: 1) promotions like flyers and coupons; 2) product descriptions; 4) product search; 5) product location; and 6) payment.

When a customer starts her shopping, she makes a shopping list of products or product categories. This information is used to personalize promotions. While the customer is inside the store, the shopping assistant receives promotions and product descriptions which correspond to her shopping list. She can also use the product search to get information about a specific product. To help her find a specific product, the mobile shopping assistant will display the floor plan of the store indicating the customer's current location, the location of the product and the shortest route in between. During shopping she can also add products to her virtual shopping cart. When the shopping is done, she can avoid waiting at the counter by using the Payment Web Service.

3. INTEGRATION ARCHITECTURE

The mobile shopping assistant is based on the architecture shown in Figure 1. The client submits a Web Service request which is sent to an asynchronous caller queue. If there is an available connection channel, the request is sent directly to the server. Otherwise, the request waits in the queue until any channel becomes available. On the server side, a broker

Copyright is held by the author/owner(s).
WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

[1]In this paper, female gender is used when talking about the customer.

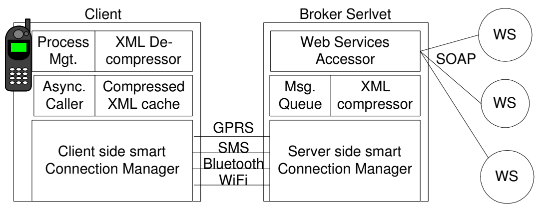

Figure 1: General Architecture

receives the client's request, puts it into a message queue and transfers it as a SOAP message to the corresponding Web Service. All the responses are processed by an XML compressor. The SOAP envelope is removed, and the XML-based data objects contained in the response are compressed to a byte stream using our context-dependent XML compression algorithm [3]. Then the resulting stream is transferred back to the client and stored in the compressed XML cache. When the client needs some data, it is de-compressed from the cache. This mechanism guarantees that the client application can continue working even if all connection channels are not available for a long time.

In our implementation, the XML compression mechanism is an important building block. It provides a highly efficient data exchange as well as a very compact data storage. Since the data is stored in a compressed form, the mobile phone can keep more information. This mechanism makes it possible to support longer off-line availability. In addition to the exact response to a client's request, the resulting response stream will pro-actively include information that might be related to the user's current request. Another important building block is a smart connection manager which can use all available communication channels.

3.1 Context-Dependent XML Compression

Traditional compression technologies, like Winzip [1] and XMill [2], are not suitable for mobile phones for the following reasons: 1) they do not work well for short messages because of the large data transmission overhead, and 2) mobile phones lack an efficient way to de-compress data, as well as process or store it.

In our mobile architecture, the XML file is used to represent data objects. The compression algorithm assumes that both the client and the server already know the structure definitions of the objects. Based on this shared knowledge, the de-compressor can interpret a compressed XML file even when the data structure information is not contained in the compressed file. Hence, we de-couple the structural metadata transmission from the actual data transmission. This approach drastically reduces the amount of bits transferred, especially for short messages. Furthermore, the coding schema for the compression can be optimized to get better compression ratio, because the de-compressor already has some knowledge of the metadata. Our experiments [3] show that the algorithm described above has at least 3 times better compression ratio than that of winzip.

To facilitate querying, an index of compressed objects is built, according to their XPath, on the client side. The index is created while the response stream is being cached. When a client is querying an object, the index is used to find the position of the object in the cache and decompress it.

3.2 Smart Connection Management

Compared to the desktop PCs, network connections for mobile phones are more pervasive and have more channels, although each channel might not always be available. Obviously, the connection availability can be drastically increased by merging all possible channels together. Hence, our smart connection manager merges several channels (GPRS, SMS, Bluetooth, WiFi, etc). It uses asynchronous communication to prevent the unstable network from causing a long delay and canceling the entire transmission.

When a message is sent, it is broken up into short packets to accommodate the limitations of channels. Each of the packets has a unique sequence number. The sender uses an available communication channel to send packets one by one without waiting for a response. On the receiver's end, the packets are assembled in the correct order to constitute the original message. If the receiver detects gaps between the sequence numbers, it requests to re-send the missing packets. Since messages are compressed byte streams, the number of packets of a message tends to be small. According to our experimental results, a 3742-byte message that contains 20 SAP business objects is compressed into 127 bytes which can be sent in one packet.

3.3 Convergent Input Modes

Although the key pad of a mobile phone is not as convenient for data input compared to a PC keyboard, mobile phones have more input modes, which in turn can extend the usage of Web Services. In the mobile shopping assistant, we use three various mobile phone input modes in addition to the key pad – camera, voice and Bluetooth. Using the camera, we implemented bar-code/sema-code scanner and image-based product search. Using the voice input, we implemented voice-based product search. Using the Bluetooth signal received from different transmitters [4], we implemented location-based navigation assistance.

4. CONCLUSION

The proposed architecture is our initial effort to make it possible for mobile phone applications and Web Services to work together. There are still many open questions in this architecture. In particular, research is underway on how to make the local cache management more efficient, how to pre-fetch data from Web Services, how to guarantee data consistency and how to solve data conflicts. In the future we also intend to turn the proposed architecture into a generic library, providing developers of mobile applications with an easy access to Web Services.

5. REFERENCES

[1] Winzip, last visit in Jan. 2007. http://www.winzip.com/.

[2] H. Liefke and D. Suciu. XMill: An efficient compressor for XML data. In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, pages 153–164, 2000.

[3] Y. Natchetoi, H. Wu, and G. Babin. A context-dependent xml compression approach to enable business applications on mobile devices. submitted to Euro-Par. 2007.

[4] M. Rodriguez, J. P. Pece, and C. J. Escudero. In-building location using bluetooth. Technical report, Universidade da Coruna, 2005.