
Challenges of Building Web Services on Mobile Devices

Rich Rollman

Abstract

Two of the hottest technology areas today are web services and mobile computing.

Web services and XML technologies are making it easier to exchange data with your partners, customers, and vendors no matter what platform, operating system, network topology, or application they use. Web services define the cross-platform, transport-independent protocols used to exchange XML messages while XML provides an extensible data representation which is easily parsed, read, and manipulated. Web services increases your data's value by allowing it to be exchanged with a greater number of people.

Mobile computing is taking the world by storm. Mobile devices have outsold PCs 3 to 1 for the last several years, with application-enabled devices outselling all PCs combined for the first time in 2003. This trend is expected to continue and accelerate in the future. An amazing and rapidly growing number of users now carry application-enabled devices in their pockets everywhere they go. The potential to leverage these omni-present, connected devices to easily access and share any data from anywhere via web services is phenomenal. However, there are many challenges that must be overcome to extend the reach of web services to mass market devices.

This paper presents the challenges you'll need to overcome to build mobile applications that use XML and web services.

Table of Contents

1. The Need for Web Services on Mobile Devices	3
2. Characteristics of Mobile Device	3
3. The Challenges of Web Services on Mobile Devices	3
3.1. Application and Memory Size	4
3.2. Processing Power, Battery Life and Heat	4
3.3. Network Speed and Latency	4
4. What is Needed	5
5. Summary	5
6. About AgileDelta	5

1. The Need for Web Services on Mobile Devices

Imagine a world in which you can access any data you want, using any device available, from any location. This is the promise of mobile computing.

Mobile computing is taking the world by storm. The wide availability of digital communication networks is changing the way people access information. Mobile devices such as cell phones, smart phones, and PDAs now have the ability to run sophisticated applications that exchange information over the Internet from a large and ever increasing number of locations. Application-enabled mobile devices now outsell personal computers by a margin of 3 to 1. The ubiquitous deployment of these devices is fueling a demand for access to more information than ever before.

While the mobile computing world has been taking shape, information access in the enterprise has been focused on making it easier to exchange data. Inspired by HTML's ease of use, XML is now the standard, platform neutral, data representation of choice. Using XML, messages can be constructed that are easily understood by an enterprise's partners, customers, and vendors. Web services evolved out of the need to exchange XML data by sending messages between different software applications running on a variety of computing platforms. As the adoption of XML and web services continues, more data is available than ever before.

It easily follows that combining web services with mobile applications would make more information available to more people at more locations than ever before. While this sounds simple in principle, implementation can be difficult unless you can overcome a number of challenges. Below, we outline these challenges and what is needed to address them.

2. Characteristics of Mobile Device

To understand the challenges which must be overcome to use web services on mobile devices, one must first understand the characteristics of different mobile devices and how these devices differ from enterprise computing platforms (personal computers and servers) for which web services were designed.

Mobile devices are smaller, lighter, less expensive, have smaller screens and are connected to lower bandwidth networks than their larger enterprise cousins. Mobile devices vary greatly in their capabilities. In general, smaller, lighter and cheaper mobile devices have fewer processing, memory and battery resources available. At the high end, PDAs and smart phones have more memory, a relatively fast processor, and megabytes of storage. Contrast these to mass market mobile devices (cell phones) carried by the majority of people and you see a number of differentiating factors. Mass market devices have less processing power, memory and screen real estate than their high-end counterparts. But they are typically smaller, lighter, simpler to use, less expensive to purchase (often included free with a calling plan) and have longer battery life. In addition, the total number of application-enabled mass market mobile devices sold each year outnumbers PDA and smart phones by more than 10 to 1.

In addition, the main memory and computing power of mobile devices, and mass market mobile devices in particular, does not grow exponentially over time as we've come to expect from enterprise systems. In fact, for years there has been almost no increase in the main memory and computing power of mass market mobile devices. This is partly due to the fact that mass market devices are very cost sensitive and partly due to the fact that Moore's law does not apply to battery powered devices, where increases in bandwidth, processing power, memory or screen real estate often translate to larger, heavier, more expensive devices with shorter battery life. Additionally, consumer features like cameras, MP3 players, video, Bluetooth, and other technologies compete for available battery life and device manufacturing costs.

3. The Challenges of Web Services on Mobile Devices

Keeping in mind the differences in platform previously discussed, let's look at the challenges one will need to overcome to run web services on mobile devices.

3.1. Application and Memory Size

The maximum size of an application on most mass market mobile devices is limited to 64 KBytes or smaller. The typical technology stack for implementing XML and web services stack consists of an XML parser, schema implementation, SOAP processor, some object model for manipulating the XML, and a WSDL implementation. This technology stack requires over 20 times the amount of memory available on mass market mobile devices making it impractical to use these technology stacks.

Application size limits the usability of mobile applications. Even if the web services stack could be reduced to 50K, this would leave only a paltry 14K bytes for the application's user interface, data manipulation, and data validation logic. Attempts to build smaller mobile web service implementations have been problematic as they typically remove features required by a large percentage of web services. Removing features creates an incompatibility between the mobile devices and data sources which employ the removed features or additional complexity for data providers to support different web services implementations or a reduced feature set. Smaller web service implementations also tend to push more of the tedious, complex programming problems up to the developer. This reduces the size of the technology stack but makes developing mobile applications more difficult, time consuming and error prone. And although smaller, these implementations still require a disproportionate amount of the mobile device's resources leaving little to no room for application features.

3.2. Processing Power, Battery Life and Heat

Processing power on mobile devices, and particularly mass market mobile devices, is typically constrained by the battery life, cost and the thermal properties of the small device. Faster processors, larger amounts of RAM, and larger color screens all require additional power, lowering the battery life of the device and/or increasing its size and weight to accommodate a larger battery. In addition, they increase the cost of the device and the amount of heat generated by the device. Heat dissipation is a serious problem for small, mobile devices which are often stored in poorly ventilated spaces (e.g., a pants pocket) and do not have the surplus battery power to accommodate active heat dissipation mechanisms like fans. In recent years there have been several documented cases in reputable news outlets of cell phone batteries exploding because they have overheated. For these reasons, there has been almost no increase in the processing power or main memory of mass market mobile devices for several years. This is due to the fact that mass market devices are very cost sensitive requiring manufacturers to balance cost and functionality (e.g. on-board camera, MP3 player), the fact that Moore's law does not apply to battery powered devices and the thermal properties of small mobile devices.

Web services make striking a balance between processing power and battery life difficult. The web services specifications were built with a large degree of abstraction which makes them usable across a wider set of platforms. But this abstraction results in longer code paths which require more processing cycles and memory thereby reducing battery life. Reducing the amount of code generally results in a reduction in functionality which, as mentioned previously, limits the amount of information available to the applications on the device and makes applications more difficult to program.

3.3. Network Speed and Latency

Unlike enterprise and desktop systems, bandwidth on mobile devices is severely limited. Instead of tens, hundreds, or thousands of megabits per second, mobile devices on a 2.5G mobile network typically get 30-100 kilobytes per second and this bandwidth is often shared with other users of the cellular network. Network latency also is a big factor. Attributing five, ten or more seconds to each transaction for latency of the network reduces the usability of the application for the end user. Therefore, smaller, less frequent transactions will result in more usable, responsive applications.

XML is used as both the protocol (SOAP) and data format for web services. XML's extensibility and cross platform features enable web services to run on virtually any computing platform. But this flexibility is achieved by representing all data types as string and the addition of tag and attribute based markup to delineate data items. XML's format significantly increases the size of the data it represents which results in longer transmission times and increased processor cycles to run the radio on a mobile device.

4. What is Needed

To build web services applications on mass market mobile devices, solutions are needed which significantly reduce both the size of the XML and web services stack and the size of the data transmitted to and from the device.

Reducing the size of the web services stack reduces memory footprint, conserves processor cycles and increases battery life. This leaves more room for application logic and user interface code which programmers can use to build more compelling applications for their users. Look for solutions that provide a complete set of web services features to ensure that applications have access to all available web services.

Reducing the size of the XML and other data transmitted within web services transactions can reduce network bandwidth requirements and increase user response time. Commercial solutions exist which can reduce the size of the XML up to 100 times and a standards effort is currently underway at the W3C to build a binary XML standard for this purpose.

5. Summary

Web services are providing access to more information than ever before using standard, cross platform formats and protocols. Mass market mobile devices are widely deployed and provide people the capability to communicate from more locations than ever before. Synergistically combining the technologies requires reducing the size of the web services stack and decreasing network bandwidth requirements of XML to build more compelling applications. These applications increase the value of the device and the information supplied by the web service by making more information available to more people when and where they need it.

Join us at XML 2005 to learn some techniques you can use to build web services for mobile devices.

6. About AgileDelta

AgileDelta, Inc. [<http://www.agiledelta.com/>] provides software for extending XML and web services to devices on the edge of networks, including mobile phones. AgileDelta's **Efficient XML** [<http://agiledelta.com/products.htm#Efficient%20XML%20Encoding>] is a lossless implementation of binary XML which significantly reduces the size of XML data and **Mobile Information Client (MIC)** [<http://agiledelta.com/products.htm#Mobile%20Web%20Services>] is an intelligent compiler framework for building compelling mobile web services and occasionally connected applications on mobile devices. For more information on AgileDelta, Efficient XML, and MIC visit our web site at <http://www.agiledelta.com>.

Biography

Rich Rollman

[AgileDelta, Inc.](http://www.agiledelta.com/) [http://www.agiledelta.com/]

15400 SE 30th Place, Suite 201

Bellevue

Washington

98007

United States of America

Rich Rollman is a founder and Vice President of Engineering and Product Development at AgileDelta, Inc., an independent software vendor that supplies XML and Web Services infrastructure software for use on mobile computing platforms. Prior to joining AgileDelta, Mr. Rollman was founder and CEO of DataConcert, a startup that built E-Business solutions using XML and Web Services to link enterprises with their small and mid-size trading partners. At DataConcert, he was responsible for the design, development, marketing, and sales of DataConcert's E-Business solutions. Mr. Rollman has been working on XML technologies since their inception. He was the Group Program Manager for Microsoft's core XML technologies (MS XML) and SQL Server's XML functionality (SQLXML). He later led product management at Crossgain Corporation where he built XML-centric programming models for web services. Mr. Rollman has 20 years experience building Internet and client-server software on a variety of computing platforms. He holds a Bachelor's degree in Computer Science and a Master's degree in Electrical Engineering.