

Learning the Lessons of WAP: A Case Study in WAP Development

Rob Machin, Concise Group Ltd

This paper will relate the experiences of a programming team that, in two months, went from knowing relatively little about WAP to developing a prototype application for an international investment bank.

It summarizes the experiences of, and lessons learned by, the team during this intensive period, emphasizing in particular the differences between delivering applications to mobile devices and desktop PCs. We will focus on what the developers considered when designing and building a prototype application for the WAP platform.

The presentation includes:

- An analysis of the available tools that were considered
- Code snippets, and an explanation of the development areas of the prototype application
- Examples of the prototype's interface on three different devices

History

In February 2000, Concise agreed to build a prototype WAP application for an international investment bank based in the City of London. The need was for a functional application that added value to the customer experience.

The bank was motivated to develop a WAP prototype for a number of reasons. Firstly, it was important for the IT function within the bank to gain exposure and understanding of the technology (and system integration issues) that would be faced as we move towards a pervasive computing model. Secondly, it was felt that a certain amount of speculative research into mobile technology was important if the bank was to be seen as an innovator in the marketplace.

The prototype delivered a WAP front-end to an existing *global-chat* system. The system allows traders to exchange market information and news on stocks and bank affairs. The application is critical to the bank, in that it facilitates the communication of market-related information between key business users.

This prototype will be developed into a full-blown application, critical to the bank's business, which will then be rolled out to all the bank's major trading centers, including London and New York

Another early Concise prototype, designed for demonstration purposes, enabled access to stock information, providing details on stocks, funds and client accounts.

We have since delivered prototypes for other key clients in the investment and private banking sectors: for delivering fund and account information for fund and asset management, and a system combining Reuters and analyst information.

Concise's WAP pilot project objectives were to:

- Identify an appropriate set of tools and platforms for delivering WAP applications
- Gain knowledge of the technologies involved in developing WAP applications
- Deliver a usable and robust WAP application that fulfilled the client's functional specifications

The project began on 7th February, starting with an investigation into tools and mobile devices. By the 17th March, an initial version of the application was demonstrated at the JSIG WAP Forum, held at IBM's South Bank Centre in London.

Since then, our focus has moved to the architecture of an extensible framework that allows WAP solutions to be delivered to multiple client devices, using an XML-based content delivery system.

This paper summarizes the experiences and conclusions of the WAP team after the intensive initial two months of the WAP development program.

Learning the Lessons of WAP

What is the purpose of a prototype?

"...you often don't really understand the problem until after the first time you implement a solution. The second time, maybe you know enough to do it right."

Eric Raymond, *The Cathedral and the Bazaar*

Overview: Project Model and Timeline

Time	Action
Week 1	Plan of architecture and investigation of technologies
Week 2	Obtaining and configuring application server and WAP gateway
Week 3	Coding EJBs, deploying and constructing servlet portion of prototype
Week 4	Coding example WML pages and connecting with Nokia 7110s
Week 5	Using LDAP for user authentication, and assembling the application
Week 6	Interface rethink and presentational issues
Week 7	Redesign of content delivery to use XML started; session management and request leasing considered
Week 8	Security issues addressed and development of further prototypes started

The Types of Tool Considered

There were many tools that we needed to select for our task — and of course, many different choices for each tool. The required tools were:

- A web server
- An application server
- A WAP gateway
- The fundamental distributed architecture:
 - i. CORBA
 - ii. COM
 - iii. EJB
- The programming tool

- Other tools:
 - i. Database
 - ii. Directory server
 - iii. WAP phone

The Products Chosen

When we had considered all the options, the final choices we made were:

Software Development Tools

- Nokia WAP server 1.0
- Nokia WAP toolkit 1.3 beta.
- Weblogic 4.51 application server and provided web server
- Netscape Directory Server 4.1 (LDAP)
- Microsoft SQL Server 7.0
- Sun Java Development Kit 1.2.2; JNDI 1.2; JSDK 2.1; J2SDKEE 1.2

Reasons for Product Choices

We made our product choices under a number of differing demands and influences:

- We needed to adopt tools that were available in the marketplace at the time the project commenced
- We needed to conform to the security requirements that represented an acceptable level of risk to the bank
- We preferred tools that were familiar to the assembled development team in order to facilitate rapid adoption and immediate productivity
- We needed to integrate with the bank's existing technological infrastructure

Choice of Mobile Device

The Nokia 7110 — one of the first WAP-enabled phones available to the UK mass market — was chosen as the initial device for deployment of our pilot application.

Server Setup

For the purpose of developing the prototype, we decided to install the web server, Java application server, database, directory server and WAP gateway on a single machine.

The server machine specification was NT Server 4.0, service pack 5, running on a Pentium III 500MHz with 256Mb RAM. The server was initially specified to handle at most five concurrent users for the first phase of the prototype.

Nokia WAP Gateway Server

We chose the Nokia WAP gateway, since it was clear it would work with the Nokia 7110 handsets, the WAP device chosen for the pilot phase rollout. Another critical reason for the choice of Nokia was that at the time the project started, it was one of the few WAP gateways commercially available that could be securely installed within an intranet environment — unlike the Phone.com offering, for example.

No problems were experienced with installing and configuring the gateway, which has proven to be robust. The gateway server is responsible for encoding and compressing the WML that is sent to the mobile device.

See <http://www.nokia.com> for information on downloading the server and toolkit.

Nokia Toolkit

The Nokia toolkit provides emulation of the Nokia 7110. Although there is no debugger for WML syntax errors (and very little help from the error messages), the simulator for the 7110 does provide a realistic and accurate environment.

The Challenge of Multi-channel Delivery

Current web application vendors can rely on their users to download the most recent version of their web browser when necessary. At present, therefore, designing and presenting content for the Web is primarily done in HTML, and it is usually aimed at the most recent generation of browser technology for the personal computer: Internet Explorer from Microsoft, and Netscape's Navigator.

The challenge now, and in the future, is in delivering meaningful content to a wide range of mobile communication devices, with a spread of computational and presentational capabilities — any implicit assumptions about the form factor of the target device must be abandoned. WAP can potentially be used to accommodate the range of form factors that will be present in the next generation of personal communicators connecting to Internet-based systems.

Integration with Existing Web Architectures

The current generation of web applications must be reinvented to succeed in the mobile market. The existing server-side structures designed for HTML are here to stay, so they must be re-engineered to play their part in the mobile Internet.

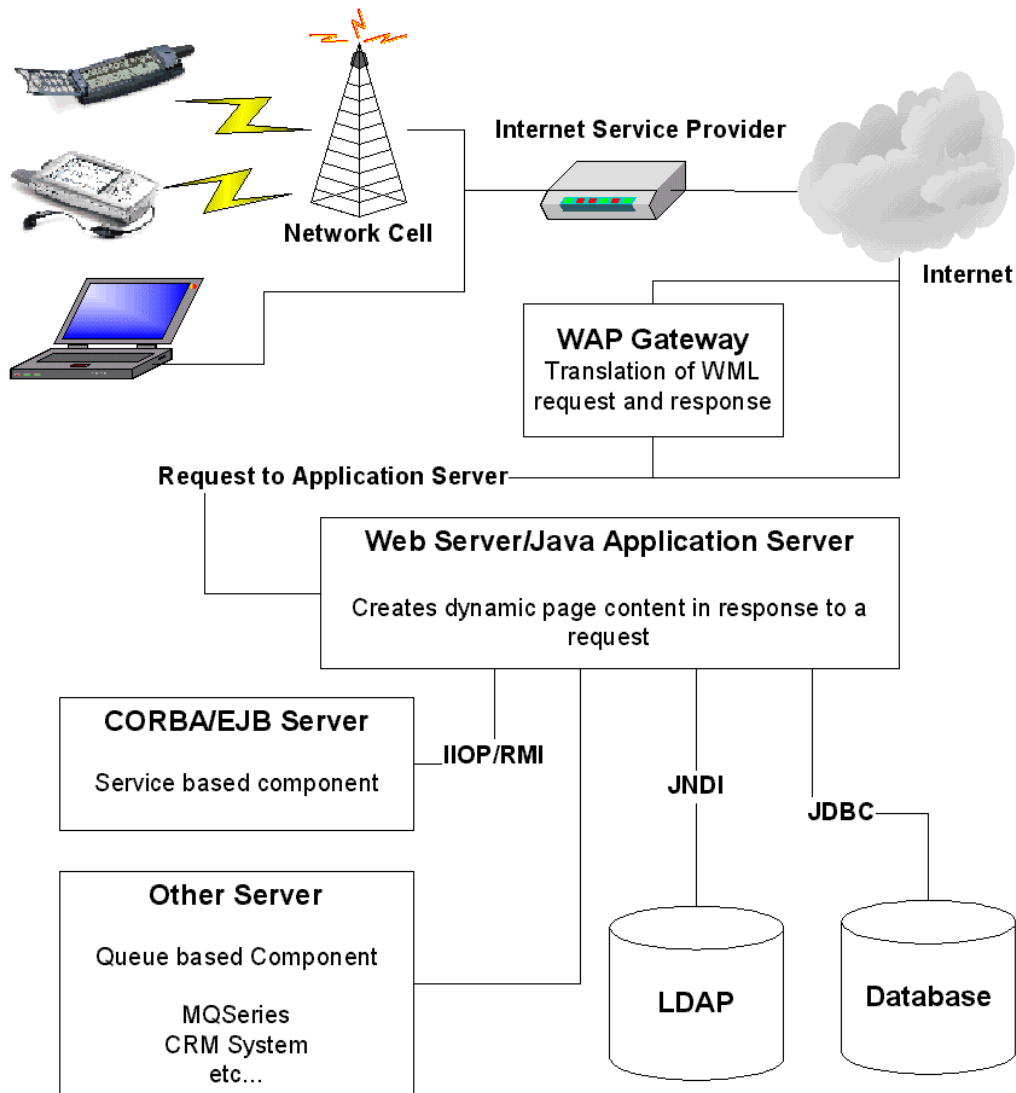
Many of our clients have existing server-side business models that can be readily adapted to deliver content to WAP-enabled devices.

Application Architecture

Our application architecture is based around the Java 2 Enterprise Edition (J2EE) platform. It consists of an n-tier, distributed computing model that can provide scalable and robust deployment of e/m-commerce systems that can be accessed securely over the Internet.

We anticipated the need for our applications to access a wide variety of backend systems. Building a *service-oriented architecture* around existing IT infrastructures will be a critical factor in the success of many web-focused systems. This architecture will allow a logical and functional (rather than physical) view of system services, increasing the ease of component-based development and reducing the time to market of new systems, by leveraging the power of legacy development.

The following diagram represents the basics of our architecture:



Services

The following section contains a list of the services that we wanted to deliver to users of WAP devices. While not all were required as part of the *global-chat* and *stock information* systems, they were investigated as part of our ongoing review of the requirements and deployment of WAP-based applications.

Session Management

Effective management of server resources, and the handling and security of user sessions, is an area that is poorly handled in many traditional CGI- and servlet-based systems. We wanted to develop a session management mechanism that would leverage the functionality of the Java `HTTPSession` object, and the power inherent in the JINI leasing model.

Security

Security constitutes one of the primary requirements in any business information system; this is particularly true of the banking and finance sector, and was something that our client was extremely conscious of.

We are currently using Netscape Directory Server to implement basic authentication and to retrieve user information. Security over the wire can be implemented using

Wireless Transport Layer Security (WTLS); WTLS is the wireless equivalent of Secure Socket Layer (SSL) also known as Transaction Layer Security (TLS).

Directory Services

Directory services, particularly LDAP, are enjoying widespread adoption at the moment, and have an important and relevant role to play in e/m-commerce. With the small screen size of most WAP devices, user configurations and preferences play a more significant role than on the larger screens of the wired Web. Delivering meaningful content to an individual using a particular device needs the services of a powerful mechanism to retrieve configuration information quickly — this a role ideally suited to LDAP.

Messaging Capabilities

Messaging and e-mail capabilities are at the heart of successful e/m-commerce. The adoption of SMS and its success in the marketplace provide a useful indicator for the predicted adoption of the more advanced capabilities of e-mail on mobile communication devices.

Interface Design Issues

Moving from HTML 4.0 to WML 1.1 was a real culture shock. Once the back-end architecture was in place, the main challenge we faced was that of delivering usable applications through a device with a small screen and limited bandwidth.

Interactivity

The growth of the Web was partly due to ordinary people being given the ability to create interactive content.

"...my definition of interactive includes not just the ability to choose (the content), but also the ability to create. We should not only be able to follow links, but create them — between all sorts of media."

Tim Berners-Lee, *Weaving the Web*

While current WAP-enabled mobile devices have little or no multimedia capability, the principle remains the same: ease of use and the ability of the user to direct and control their online experience is crucial to the success of WAP applications.

Usability and Flexibility Mean Allowing the User to Know Best

The challenge we faced with WML was to allow individuals the ability to manipulate, link and create content through their WAP devices as easily and creatively as they were able to in the mid-90s with HTML on the Web.

From our experience in writing WML, and the feedback we received from our test users, we knew that more attention had to be paid to the usability of our menu layout and overall navigation design.

Applications must allow the user to choose the functionality they require from the business functions available; preconceptions about layout, structure, and content must be kept to a minimum. This is especially true for devices with limited bandwidth and computational resources, where redundant functionality will cost the user both unnecessary time delays and expense.

This is the challenge that must be met in order to accelerate the acceptance of WAP devices; it will require a new way of thinking about the presentation of information and the input of data through non-computer terminal devices.

Size Does Matter

Even a basic computer monitor can display 800 by 600 pixels, and a single line of text can accommodate 100 or more readable characters. We also have color and a wide variety of different styles, typefaces and graphics at our disposal in order to make the browsing experience both easy and pleasurable for the user.

The screen size on the Nokia 7110 mobile phone is 96 by 44 pixels — less than one-hundredth of the area! It can only fit around 14 characters on one line of text.

Our initial interface was very much driven by the structure of the basic functional requirements. After implementing a rather clunky UI first time round, we realized that to design WAP applications successfully, we had to rethink things completely. It was necessary to abandon all preconceptions about functionality in order to take advantage of WML, and avoid its limitations.

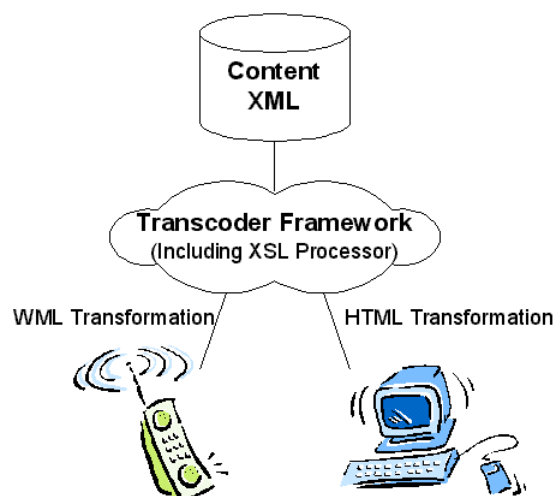
Our 'second take' at the user interface design considerably improved the user experience, and increased our awareness and understanding of the issues involved. The key areas we targeted were:

- Reduction of the amount of user input — it is slow and cumbersome on a phone
- Simplification of the menu structure
- Giving easy access to the main menu, to stop the user from feeling lost
- Consistent naming of menu options
- Reassuring user with visual clues (such as the synchronization of menu options and the titles of screens)
- Providing the ability to undo any choice easily

Navigation is more difficult with a small screen, so the menus have to be extremely easy to navigate, and the structures as shallow as possible. Clicks/selections must be kept to an absolute minimum. Simplicity and brevity are the name of the game.

Web Site Design in the XML Era

The advent of XML has allowed programmers to handle the development of web pages in a different way, especially when utilizing the potential of XSLT stylesheet transformations.



Designing for Growth; Designing for Change

A major challenge of developing for mobile devices is dealing with the difference between the relatively static *content* format and rapidly developing *presentation* formats. The XML/XSL combination can help you to deal with multiple client devices, from the Nokia 7110 to the next generation of PDAs.

Our solution was to use XML for all our content, allowing us to use industry standard grammars as they appear. We also used stylesheets to transform the XML into the optimum WML for the particular client device. This allows us to deal easily with new devices as they appear, and to maintain our services to legacy devices with the minimum of coding overhead.

Have we met our Goals?

We wanted to write scalable, secure, WAP-based, distributed applications that delivered business value.

We aimed to understand how to structure content with WML, and how to implement security using WTLS and LDAP. Another thing that was important to us was gaining a general understanding of the WAP industry and market.

Our ultimate aim was to deliver to the user an intuitive UI with access to dynamic, timely and meaningful content that was sufficiently secure to protect the confidentiality of the information.

How we Fared, and what we Delivered after Two Months

We initially underestimated the challenge of designing for a device with such limited presentational capabilities. Initial menus proved difficult for users to navigate, and were replaced by a simple, clear design with shallow menu structures and the minimum of user clicks.

We used LDAP/JNDI to authenticate users, but we did not implement end-to-end security or use WTLS in the first phase. We did, however, manage to deliver a working prototype that accessed the business information, and generated dynamic content based on information stored in a back-end data store.

Conclusions

The most important elements in a successful WML application are:

- A simple-to-navigate, menu-driven interface.
- Designing for change: the initial pilot generates Nokia 7110-optimized WML directly, but in the next stage we will use XML and stylesheets to optimize access to our application from any device, from mobile phone to mainframe.
- Open standards compliance

The technologies we plan to use to implement our first generation products include Concise's nascent technology program — looking at the future and being one step ahead:

- Scalable Java backend — Servlets, EJBs
- Watertight security with LDAP and SSL/TSL
- Flexible content with XML and XSL
- WML to the limit — delivery to WAP clients using optimal WML
- Industrial strength database technology

We aim to advance our development effort both through the partnerships we have developed with industry leaders, and by remaining aware of and involved in new and emerging technologies in the e/m-commerce arena. Partners currently include:

- Sun Microsystems
- BEA
- Nokia
- Psion

BEA will be including the Nokia WAP gateway in future releases of their application server offering, *Weblogic*.

After eight weeks of intensive WAP development, we remained convinced of the importance of this technology and continue to look forward to the next stage in its evolution.