WAP: Untangling the Wireless Standard

Network Computing

November 27, 2000
By Peter Rysavy

Wireless data is hotter than ever, especially with new data options for cellular phones, new mobile platforms, Internet portals with mobile content and the prevailing desire to remain connected as we roam from one place to another.

A key technology in this wireless maelstrom is WAP (Wireless Application Protocol). WAP and its precursor, HDML (Handheld Device Markup Language), make it possible to reliably and efficiently communicate data over wireless WANs, which are notorious for high error rates, slow speeds, high usage costs and intermittent connections.

The WAP Forum, founded by Ericsson, Motorola, Nokia and Phone. com, controls the WAP specifications and boasts hundreds of members and considerable worldwide market momentum. Users typically interact with WAP content using microbrowsers built into cell phones, with screen sizes ranging from four to 11 lines, each line containing 12 to 16 characters. If WAP takes off, every mobile telephone could come with WAP built in within two years. And with hundreds of millions of cell phones, optimists predict massive production of mobile-oriented content and applications. The applications that make the most sense for WAP will be transactional in nature, compared with general Web surfing, because of form and bandwidth constraints. People will be checking the weather, buying theater tickets, rescheduling flights and ordering lattes using WAP.

WAP browsers are becoming available for other handheld platforms as well. Moreover, WAP does not limit screen size, which eventually will increase. Vendors are offering a growing number of tools for developing WAP applications, and there are also off-the-shelf solutions that let organizations extend standard business applications, such as Microsoft Exchange and Lotus Notes.

Maybe you have some intranet content you would like to deliver to your mobile work force. Examples include helpdesk automation, pricing information, telephone databases, messaging and calendar management. Or maybe you have a great idea for horizontal-market content you would like to sell to mobile users at large, such as rebooking people on alternative flights when original flights are cancelled. To do this, you'll need to be familiar with WAP's architecture, the steps involved in developing applications and some of WAP's problems.

WAP in Perspective

Wireless applications must match users' needs. There is no reason to deliver information over a wireless network if users can wait until they return to their offices to download it, and there is no point in designing applications that are frustrating to use because of the mobile device's physical limitations or bandwidth limitations.

The founders of WAP designed the protocol to address the bandwidth and latency issues of wireless networks. They also wanted to optimize menu-driven applications for microbrowsers operating on mobile telephones. Another goal was to provide a mechanism by which operators can control how users access different content and applications, paving the way for value-added objectives. WAP lets developers host content and applications on existing Web services for additional revenues. The WAP Forum has met

these objectives successfully. WAP lets developers host content and applications on existing Web servers that mobile users can access regardless of device and wireless connection.

But there are severe limitations, such as the tiny amount of display space and the difficulty of entering text using a telephone keypad. WAP also requires that developers format the content using WML (Wireless Markup Language), which, though based on XML, involves new syntax.

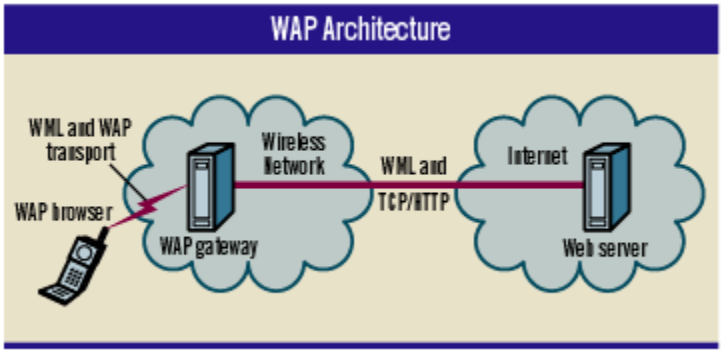
Because cellular operators handle the WAP gateways through which WAP content passes, broad-market developers must negotiate with carriers to have the developers' sites listed on the phones' home pages. Finally, WAP is a set of proprietary protocols controlled by the WAP Forum and not subject to the open review process of Internet standards. Although there are efforts under way to eventually converge WAP with World Wide Web Consortium (W3C) standards, this could take years.

Keep in mind also that the speed of wireless networks is increasing. By the end of 2001, access speeds ranging from 28 Kbps to 128 Kbps will be common, at which point the standard HTML becomes feasible over wireless connections.

Nevertheless, WAP can make sense for carefully targeted applications. The key is to have small amounts of content that have high real-time value and require minimum user input.

WAP is not the only game in town. Major U.S. carriers still use WAP's precursor, HDML. In Japan, more than 10 million users have subscribed to a service called i-mode, which offers capabilities similar to WAP's but is based on a subset of HTML, called compact HTML. NTT DoCoMo, which created i-mode, is trying to export i-mode to other countries, and in fact a future version of WAP may support i-mode content. This is a quickly evolving area that requires close monitoring. Don't get involved with WAP unless you have a good feel for the market and have an appropriate application. WAP is not a cure-all, and skeptics consider it an interim technology at best.

WAP Architecture



More than just a set of protocols, WAP is an entire architecture (see "WAP Architecture," at right). The key components are the microbrowser in the mobile device (usually a mobile telephone, but browsers for handhelds are also coming) and the WAP gateway (also called the WAP proxy), which usually resides in the carrier's network. There is also the Web server that hosts the WAP content.

WAP's page model uses the concept of cards and decks. One card typically presents one screen of information. A deck is a group comprising one or more cards with links between them, allowing local navigation of content. When a device requests a deck of information, the request travels to the WAP gateway, which forwards the request to the server. The server responds by sending content in WML (or HDML) format to the WAP gateway, which forwards the response to the user.

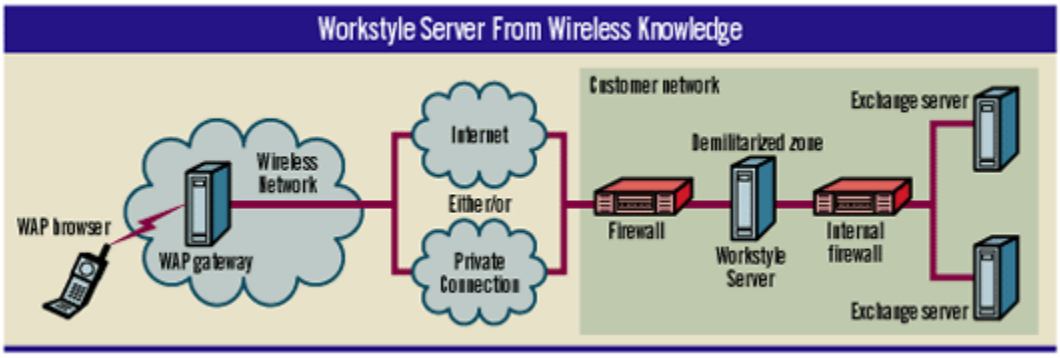
Why have a gateway? For reliable wireless communications, the gateway and mobile browser use specialized transport protocols at Layer 4 of the protocol stack. For example, WAP specifies a reliable transaction protocol that is optimized for wireless communications and is more efficient than protocols such as TCP. WAP's transaction protocol does not use the three-step handshake used by TCP when it establishes a connection. As for Layer 3 of the wireless connection, some networks use IP while others

use proprietary protocols. It is even possible to use the SMS (short message service) for transporting WML in digital cellular networks.

The advantage of WAP is that the wireless transport is independent of the content. In other words, developers can theoretically produce WAP applications without worrying about details of individual wireless networks or devices. The gateway also benefits carriers, as it controls access to services.

The default home page on the mobile telephone supplied by the carrier points to its gateway, and so the carrier determines what sites are listed on this all-important initial screen. Although users can enter any URL, this is awkward given the limited keypad.

Communication between the gateway and the Web server is based on standard TCP and HTTP. Because the application resides on the server, no application software needs to be installed on the mobile device. For security, WAP provides encryption and authentication through WTLS (Wireless Transport Layer Security), a protocol that operates between the browser and the gateway. WTLS is based on TLS (Transport Layer Security) and SSL (Secure Sockets Layer) but accommodates limitations of wireless connections, such as higher latency. Actual encryption options include DES and 3DES. Between the gateway and the Web server, standard Internet security protocols, such as SSL or TLS, can be used.



Off the Shelf

Just as you can do with HTML, you can roll your own or buy an off-the-shelf solution. One approach with off-the-shelf solutions is to take content that might otherwise employ HTML or other client rendering methods but deliver them in WML format. An example of this is a product called Workstyle Server from Wireless Knowledge, in San Diego, a joint venture between Qualcomm and Microsoft (see "Workstyle Server From Wireless Knowledge," above). You install this server in your network between your Microsoft Exchange servers and an external network, such as the Internet, that provides a connection to the wireless carrier. The firewalls shown are optional but a good idea, particularly the external one if you employ a connection to the Internet. The Workstyle Server acts as a proxy on behalf of the mobile user, accessing the Exchange server for mail and calendar information and delivering it to the WAP gateway in WML or HDML format. Lotus Development Corp. is developing a comparable capability with its Mobile Services for Domino product. Expect similar features for other enterprise packages, such as Apple Computer's WebObjects, BEA Systems' WebLogic, IBM's WebSphere and PeopleSoft's Mobile eStore.

You also can outsource the mobile-content hosting function to an increasing number of wireless ASPs (application service providers), such as Broadbeam Corp., InfoSpace and OracleMobile. These service providers not only host applications but can also assist you with the integration of your solution.

Roll Your Own

If you're going to develop your own application, you will need to produce WML or HDML code. HDML has capabilities similar to WML's but is controlled by Phone.com. Phone.com created HDML before WAP even existed, and HDML has continued to evolve in parallel with WAP. In the United States, you need to consider both formats, as AT&T Wireless Services, Sprint PCS and Verizon support HDML, but Nextel Communications and VoiceStream Wireless support WAP. Much of the information here on WML also applies to HDML. WML, which is based on XML, specifies the format and presentation of text, hierarchy of pages and links. There is a version of JavaScript, called WMLScript, that allows local logic execution.

What about converting existing HTML? There are conversion tools available but don't expect any magical process. There is no easy way of effectively condensing rich content to a tiny screen. You really need to design the WML content, taking into account the user interface. You can develop new content in XML and then use extensible style sheets to render the content on different platforms.

There are several WML editors, including online editors, available (see Web Links, at left, for information on AnywhereYouGo. com). There are also several emulators you can use. These allow you to interact with WML content using graphical representations of mobile cell phones right on your large-screen monitor. The content can be on your hard drive or on a separate Web server. Companies such as Nokia and Phone. com also offer SDKs for developing HDML/WML content.

Web Links:

- WAP Forum

  (www. WAPforum.org) offers specifications and background material.

- AnywhereYouGo.com (www.AnywhereYouGo.com)

  offers testing and development support for WAP, and a comprehensive listing of tools and third-party providers. This is a good site to get you started.

- The WAP founders

  -- Ericsson

The following short example of WML displays a simple screen saying, "Press the Accept button for the next screen."

```
<wml>
<card>
<do type="accept">
<go href="#nextcard"/>
</do>
<p>
Press the Accept button for the next screen
</p>
</card>
```

(www.ericsson.com), Nokia (www.nokia.com), Motorola (www.motorola. com) and Phone.com (www.phone.com) -- all have extensive information about WAP.

The do statement associates a means of invoking an action (pressing the Accept button) to an action, namely going to the next screen. The go statement specifies the URL for the action. The

prints the text on the screen. The advantage of WML editors is that, much as in HTML editors, you don't need to worry about exact syntax. And if you want to elevate yourself one level, tools from companies such as AvidWireless are designed to let you specify content at a higher level of abstraction, wherein the tools format the content for the appropriate user device.

These tools free you from keeping up on all the finicky details of each target device.

WAP Shortcomings

WAP and HDML are very effective for the right applications, but the protocols--and particularly their implementations--have some problems.

First, you have to decide whether to develop your format in HDML or WAP or both. And there are many versions of HDML and WAP available, causing further complications. Multiple phones with varying capabilities also complicate the scene. Their numbers of lines of display vary, as do their graphics capabilities. To some extent, WAP automatically handles different screen sizes, but you still have to decide whether to design for the lowest common denominator. Even worse, because of ambiguities in the WAP specification, different vendors have implemented features differently, and the only way to be sure your application runs correctly is to test it with all the target devices of interest. In addition, WAP gateways from different vendors do not behave identically; therefore, because your application works on one carrier's network does not mean it will work on another carrier's network. These interoperability issues have been plaguing the WAP community, and although major interoperability testing programs have been launched, problems persist.

Security is another issue. Because WAP relies on different security protocols on the wireless links and on the wire-line links, information must be decrypted and re-encrypted at the gateway, resulting in a security solution that is not end to end. The WAP Forum intends to fix this in future versions of the specification.

Despite these difficulties, a large number of companies are developing WAP tools, testing services, hosting services, devices and applications. The industry is in a pioneering stage with a rocky road to travel, but WAP is likely to be used in many new applications.

Peter Rysavy is president of Rysavy Research, a consulting firm that helps companies research, develop and deploy communications technologies.