Published in final edited form as:
IT Prof. 2012 ; 14(3): 44–49.

MedlinePlus Mobile: Consumer Health Information On-the-Go

Loren Frant, **Brian Goldstein**, **Yue Ma**, **Dianne Sun**, and **Sarena Burgess**
US National Library of Medicine

Since 1998, the US National Library of Medicine's MedlinePlus.gov website has offered reliable, up-to-date health information for free to patients and their families and friends. It contains information on over 900 diseases, conditions, and wellness issues (for more information, see the sidebar). In 2009, with mobile Internet use on the rise, NLM began creating MedlinePlus Mobile, a mobile-optimized version of the full MedlinePlus website. Many companies offering mobile solutions, such as WebMD (www.webmd.com) and NASA (www.nasa.gov), created device-specific (iPhone and Android) apps. However, the MedlinePlus Mobile team wanted to reach as broad an audience as possible, and we didn't have the resources to build and maintain separate apps for a variety of platforms. A mobile optimized site thus presented the best way to reach the diverse audience of mobile users seeking health information.

Here, we outline MedlinePlus Mobile's development and implementation.

The Challenges

As part of the MedlinePlus Mobile team, we investigated both the low and high end of the mobile device spectrum. According to "The Worldwide State of the Mobile Web," a report published by Nielsen in 2008 (the time of the our initial research), approximately 10 percent of the mobile Web was accessed by devices that only supported Wireless Markup Language (WML)—such as the Motorola RAZR, which couldn't handle newer elements of HTML or XHTML (see Table 1).[i]

At the time of MedlinePlus Mobile's initial development, a significant percentage of mobile Web searchers used the RAZR and other devices like it. We took these capabilities into consideration during the development, which posed one of the major challenges of the project.

Ensuring MedlinePlus Mobile worked well on a variety of devices required providing users with varying views based on device capabilities. We thus needed a robust way of determining a user's device type. We also needed to provide content to low-end phones that only supported WML content.

In addition, because the full-site content had been designed for desktop and laptop viewing, resizing many of the MedlinePlus assets was essential to making them easily viewable on any mobile device.

Because MedlinePlus Mobile contains a subset of the content that appears on the full site, we had to figure out how to separate the mobile content from the full-site content without drastically impacting current workflows.

Finally, we needed to accurately test MedlinePlus Mobile on a variety of devices to ensure they displayed the correct view and that browsing, navigation, and searching worked on all devices.

System Architecture

We developed an automated content management process for MedlinePlus Mobile to leverage the process already in place for MedlinePlus, resulting in no need for increase in staffing or other resources. Understanding this solution requires first understanding how the MedlinePlus system operates.

The MedlinePlus architecture involves an expansive Adobe ColdFusion input system and an overnight build process that generates a set of public HTML pages. The system maintains a set of records in an Oracle database maintained by NLM librarians through a custom input system. MedlinePlus also displays licensed content from various providers that's updated regularly. An overnight build process pulls records maintained by the MedlinePlus librarians from the database to generate a set of static HTML files for public display.

We had to preserve the performance of the ColdFusion system and Oracle database without adding complexity to existing rules and processes. So we selected a minimally invasive approach to implementing a mobile version of MedlinePlus that contained a subset of the content on the full site.

The system generates the mobile XML feeds by expanding the current build procedure that generates the MedlinePlus HTML pages. This mechanism not only ensures data integrity for both the main and mobile sites but also reuses the same MedlinePlus build procedure for the mobile feeds to preserve system performance. These XML files provide an easy way to render the content in the multiple views required for a comprehensive mobile environment. Figure 1 shows how MedlinePlus Mobile fits into the existing architecture.

To maintain the MedlinePlus Mobile data, a comprehensive XML schema describes the mobile feed data structure. At compile time, the Java Architecture for XML Binding (JAXB) un-marshals the MedlinePlus Mobile XML schema into Java classes that encapsulate the MVC model. AXB lets us easily add new types of pages and new structures with simple changes to the XML schema as opposed to large amounts of redundant Java code.

Device Detection

Being able to analyze individual users' devices on the fly was one of the unique challenges the MedlinePlus Mobile project faced in its foray into mobile development. To obtain the best possible view for users visiting the site, we needed a comprehensive system for determining each user's device capabilities dynamically upon entry into MedlinePlus Mobile. A critical element to that was identifying a reliable, trustworthy mobile device information repository.

There are many available solutions for this problem. One of the simplest involves using an alternate Cascading Style Sheet (CSS) with the media type “handheld.” Although this is useful in some situations, because MedlinePlus Mobile only displays a subset of the full MedlinePlus content, and because of the unreliability of individual browsers, this wasn't an acceptable solution.

After researching various approaches, we implemented device detection using the Wireless Universal Resource File (http://wurfl.sourceforge.net) by working with a group of Italian consultants led by Luca Passani, WURFL's inventor.

WURFL is an open source repository of information containing capabilities and features for thousands of mobile devices. It collects information into an XML file that can be easily parsed with its proprietary APIs. It was originally managed by WURFLpro and an open

source community; however, WURFL and its data and APIs are currently maintained by ScientiaMobile based in Herndon, Virginia.

WURFL is regularly updated with current device specifications, letting its users provide optimized views for all mobile devices—including new ones as they hit the market. WURFL let us confront the challenge of device detection in a comprehensive way that enables MedlinePlus Mobile to reach as many devices as possible, including low-end mobile devices. For example, WURFL tells the MedlinePlus Mobile program the specific device's screen size so it can optimize the images, font size, and other layout elements as needed.

When a user enters MedlinePlus Mobile, the WURFL repository is accessed by the application using the Java API provided by ScientiaMobile. MedlinePlus Mobile then analyzes the device capabilities it pulled from WURFL to determine the appropriate view set. MedlinePlus Mobile breaks devices into two main views: XHTML for middle to high-end mobile devices and WML for low-end devices. It also further breaks down the XHTML views to refine them for specific higher end devices, such as the iPhone, Android, and Blackberry. Once MedlinePlus Mobile selects the view for that specific device, it then renders the page content for display by Java Server Pages (JSP).

A major consideration during development was pagination. The physical limitations of many devices would require excessive scrolling to view the full content of some articles and entries on the site. We solved this issue by using WURFL to determine whether a device is high end and thus capable of displaying the full content. If not, the system breaks the article up into smaller pages so users can access the specific information they need without scrolling through hundreds of lines of text.

One final aspect of device detection was preventing mobile device users from unintentionally landing on the full site. To handle this, we chose an open source solution provided at detectmobilebrowsers.com. We installed the Apache version of the script to redirect mobile device users who access the desktop site to the mobile optimized site. To accommodate users with high-end devices who want to use the full site, there's a link back to the main site. This link sets a cookie on the user's device to avoid future redirects.

Image Handling

To give MedlinePlus Mobile users an interactive service, we had to optimize images for both low and high-end mobile devices. The solution was to resize the existing images into various sizes that would be dynamically chosen based on individual users' device capabilities.

Given the wide range of device screen sizes, we needed to consider how each image would look on the mobile site. Aesthetics, however, wasn't the sole reason to resize the images. Many lower-end devices have trouble loading large amounts of data, to the point where the page won't load at all. It's better to reduce the data load by resizing images than to remove them entirely. The best possible approach was to resize the images so they wouldn't exceed the device's natural width. This way, the user would never have to scroll horizontally to view a page.

To accomplish this, we created three image roles—full-, half-, and third-screen—which were then subdivided into five different widths to cover all possible devices. The widths chosen were 165, 228, 300, 320, and 440 pixels. For example, many encyclopedia articles on MedlinePlus link to several illustrations through a gallery of thumbnail images. In this gallery, the thumbnails might be assigned the third-screen image role, because three images could be displayed side by side on the screen. If the device's maximum width were 280

pixels, then MedlinePlus Mobile would proportionally resize the images to a width of 76 pixels.

Testing and Implementation

Testing a mobile site accurately for numerous different devices was difficult. Devices available in-house at NLM didn't cover the entire range of use cases that required testing. We needed an efficient and cost-effective way to test on a wide range of devices.

DeviceAnywhere offered the best test platform by providing Web-based access to hundreds of actual mobile devices (www.deviceanywhere.com). Using this service let us determine where the MedlinePlus mobile-optimized site needed improvement for specific devices. It also provided a way to test the mobile redirect on the MedlinePlus full site and to see exactly what MedlinePlus users would see when visiting the site.

Through rigorous testing using DeviceAnywhere, we identified most, if not all, problem areas. Most of the issues encountered related to how the different mobile browsers interpreted CSS. However, other issues related to how certain high-end phones handled the “viewport” metatag and how low-end devices can handle only a small amount of data in a single request.

NLM released MedlinePlus Mobile in January 2010 in both English (http://m.medlineplus.gov) and Spanish (http://m.medlineplus.gov/espanol). It features the summaries for over 900 diseases, conditions, and wellness topics as well as the latest health news, an illustrated medical encyclopedia, a medical dictionary with audio pronunciations, and information on thousands of prescription and over-the-counter medications.

MedlinePlus Mobile Search

MedlinePlus Mobile includes a comprehensive site search in English and Spanish, as well as the ability to look for specific health topics or medications in either language. These search implementations are powered by the Vivisimo Velocity Platform (http://vivisimo.com/technology/velocityplatform.html), the same search engine technology that powers the full MedlinePlus website and other NLM sites. Using the Vivisimo search engine in MedlinePlus Mobile ensures consistent indexing and retrieval behavior with other MedlinePlus-branded products and builds on an existing enterprise tool that already performs well in other NLM contexts.

In the Vivisimo implementations for the full MedlinePlus website, search result pages are hosted on the Vivisimo servers, and the Velocity software controls the HTML results display. To optimize the display of search results for an array of mobile devices, rather than generating static HTML results pages, we use the MedlinePlus Mobile program to dynamically generate search result pages.

In this model, MedlinePlus Mobile uses the Vivisimo search engine as a Web service. When a user enters search terms into a MedlinePlus Mobile search box, the MedlinePlus Mobile program sends a request containing the query to the Vivisimo search engine. The search engine responds to this request by delivering search results in XML format back to the MedlinePlus Mobile program. The MedlinePlus Mobile program then renders the Vivisimo XML for display by applying the same WURFL-based approach used in other content areas on the mobile site.

This approach lets us use Vivisimo features—such as relevancy ranking, spelling suggestions, synonym expansion, and keyword-in-context highlighting—while retaining the ability to tailor the display to specific device types.

Future Opportunities

At the time of development, using Java with the Spring Framework and JSP was attractive due to the strong in-house knowledge NLM already had with those technologies. With the introduction of new technologies and frameworks for building Web applications, such as Sencha (www.sencha.com) for building HTML5 applications, more efficient solutions now exist for bringing the content MedlinePlus has to offer to mobile users. In the future, these new technologies can help expand the reach of MedlinePlus Mobile and provide a more interactive experience.

NLM plans to revisit a few areas of the MedlinePlus Mobile infrastructure, possibly phasing out WML devices as mobile Internet users gain access to more robust devices. Currently, a small percentage of users accessing MedlinePlus Mobile are using devices that require WML support. Phasing out WML support would let us better focus MedlinePlus Mobile's design and user experience. In addition, discontinuing WML support would increase image-processing efficiency.

We might also be able to improve overall image handling. The current system requires a lot of overhead and storage to keep 15 versions of each image. As device resolution and technology improves across the entire mobile marketplace, we will be able to eliminate some versions.

We plan to improve back-end efficiencies to make MedlinePlus Mobile easier to maintain. Any changes made to MedlinePlus Mobile in the future will uphold or improve upon the site's quality and performance.

Finally, MedlinePlus Mobile doesn't include links to external sites, most of which aren't optimized for mobile devices. As consumer health information producers create mobile-optimized versions of their sites, we will consider adding those external links to MedlinePlus Mobile.

NLM is in the middle of a 12-month evaluation of MedlinePlus Mobile, which started 1 November 2011. As part of this evaluation, we're determining who uses MedlinePlus Mobile, what they use on the site, and their level of satisfaction. We're collecting data from Web analytics (WebTrends and Google Analytics) and gathering end-user survey data through the American Customer Satisfaction Index (www.theacsi.org). We'll use the data to make incremental improvements to the site and identify areas for further study.

Biographies

Loren Frant is the head of the Health Information Products Unit at the US National Library of Medicine. Her professional interests include Web management, systems design, and consumer health information. Frant has an MS in library and information science from the University of California, Los Angeles. Contact her at loren.frant@nih.gov.

Brian Goldstein was a software engineer for Lockheed Martin, working at the US National Library of Medicine during this project. He is now a software developer at Finra. His professional interests include Web and Java development. Goldstein received his BS in computer science from the University of Maryland, College Park. Contact him at brnsgoldstein@gmail.com.

Yue Ma is a senior software developer/analyst for Lockheed Martin, working at the US National Library of Medicine. He currently serves as a developer for the consumer health website MedlinePlus.gov. His professional interests include high performance website development, XSLT, and AJAX. Ma received his BS in applied physics from the Branch Campus of Peking University, Beijing. Contact him at yue.ma@lmco.com.

Dianne Sun is a technical project lead for MedlinePlus and MedlinePlus Mobile at the US National Library of Medicine. Her professional interests include computer IT and design. Sun received her BS in electrical engineering from the University of Maryland, College Park. Contact her at dianne.sun@nih.gov.

Sarena Burgess serves as a librarian on the MedlinePlus team in the Health Information Products Unit at the US National Library of Medicine. Her professional interests include consumer health information services and technologies. Burgess received her MS in information science from the University of Tennessee. Contact her at sarena.burgess@nih.gov.

References

i. Critical Mass: The Worldwide State of the Mobile Web. Nielsen Mobile. Jul. 2008 http://nl.nielsen.com/site/documents/nielsenmobile.pdf

MedlinePlus Mobile provides a core set of the authoritative health information found on MedlinePlus.gov, optimized for mobile devices. The authors detail the technical approach taken to create this website, from the system architecture through development and testing.

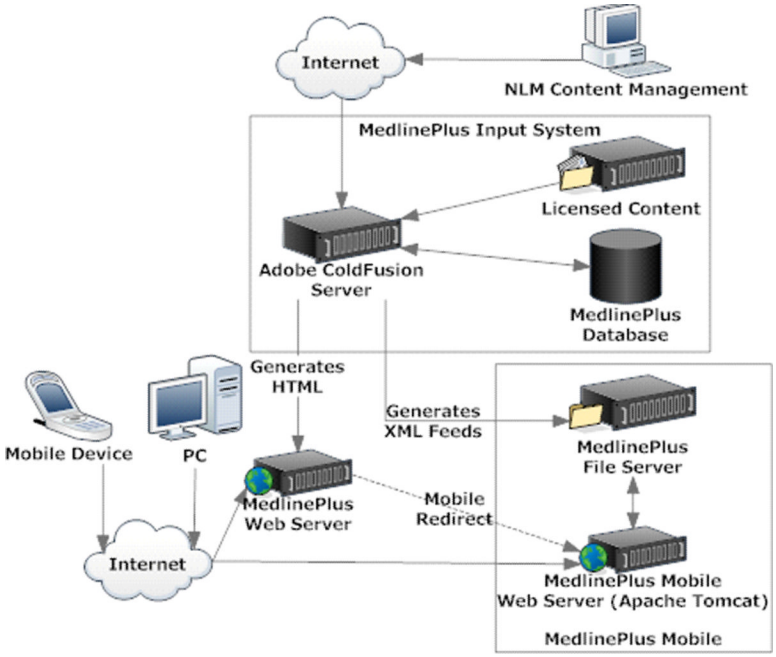

Figure 1.
MedlinePlus Mobile uses a Model-View-Controller (MVC) model.

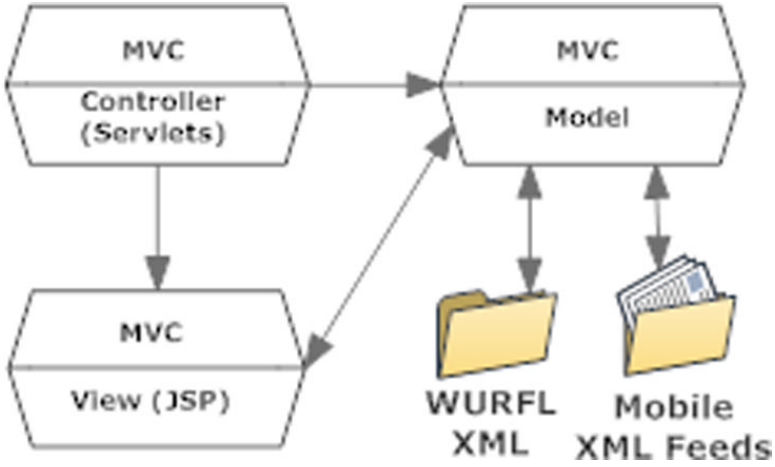

Figure 2.
The MVC model (www.springsource.org), which features its own MVC model that lets us process different views. It also handles the dynamic injection of certain runtime objects.

Table 1

Top devices of mobile Internet users in the US for the first quarter of 2008.

Device	Mobile Web access
Motorola RAZR/RAZR2	10%
Apple iPhone	4%
RIM BlackBerry 8100 series (Pearl)	2%
RIM BlackBerry 8800 series (8820, 8830)	2%
Motorola Q Series (Moto Q, 9h, 9c, 9m, QGlo)	2%