Towards the generation of mobile device markup from web pages

W. Chang & R. Kelly
Department of Computer Science, Stony Brook University, USA

Abstract

The variation among mobile devices has led to various approaches for the presentation of data. In general, these approaches work well when a specialized application is developed for a given device, but do not work as well when generally available web pages are used as the source. Various approaches have been proposed and used for the display of web pages, most recently the proposal for HTML Mobile Profile. The evolution and implementation of this standard have opened the possibility of near-universal support for mobile device web display through conversion of web pages into a form consistent with HTML Mobile Profile. RIML, a markup language developed by the Consensus project took a major step in this direction with a system that includes an enhanced markup language, a database of device characteristics, and a translation engine. It introduced a rich set of mapping concepts, but the extensions to HTML make it unrealistic to reach widespread implementation. Our approach is similar to the approach of Consensus, but instead uses existing HTML syntax to provide translation pragma. The pragma defined in the HTML guides translation engine that breaks the page into a deck-of-cards, with the appropriate links between cards. We approach the problem of transforming existing web pages into wireless content by dividing it into three separate problems: the conversion of unstructured or invalid web pages into valid pages (XHTML), the transformation of valid pages into a markup suitable for mobile devices (XHTML-MP), and finally, the formatting of the resulting documents such that they are suitable for display on wireless devices. The pragma specified in the HTML guides these steps, particularly the formatting of the document wireless display.
Keywords: XHTML, HTML, XSLT, WML, mobile devices, mobile profile, wireless web, deck of cards, content adaptation, automatic pagination.

WIT Transactions on Information and Communication Technologies, Vol 36, © 2006 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/IS060311

1 Introduction

The phenomenon of the World Wide Web (WWW) has brought about many changes to the way we live: activities such as online shopping, online bank account management, and web video conferencing have all become the norm thanks to the technologies brought about by the Internet. Likewise, the proliferation of the WWW has also brought along an array of web-enabled devices: personal digital assistants, wireless cell phones, and even hand-held video game devices, allowing ubiquitous access to the web. Unfortunately, depending on the actual hardware and software specifications, browsing the web using a mobile device is not necessarily a pleasant experience: many websites do not display correctly, or do not even display at all.

That the majority of existing web pages are invalid is a major cause of the poor mobile browsing experience. In fact, the World Wide Web Consortium (W3C) [1] Quality Assurance Interest Group suggests that approximately 99% of all web pages are invalid, and a study by Parnas [2] of the University of Bergen confirms that indeed only 0.7% of all web pages are valid. Because of the lack of processing power on most mobile devices, many mobile browsers are incapable of displaying incorrect web pages. Furthermore, it is very difficult to transform non standards-compliant web pages into other markup language formats (including mobile markup languages) consistently.

In addition to the large proportion of invalid web pages in existence, the inherent mismatch between mobile device usability and desktop computer usability also detracts the mobile browsing experience; the bulk of web pages available on the Internet are designed for web browsers on personal computers, which do not suffer from hardware constraints such as limited screen size, lack of computational power, and a different set of input methods. Therefore, even valid web pages may display incorrectly on a mobile web browser.

The Wireless Project was started to enable mobile access to the rich amount of data available on a Stony Brook University web site, and then extended to facilitate the development of mobile web applications. We propose a framework under which developers can extend their existing web content to mobile devices utilizing existing HTML (Hyper Text Markup Language) meta-data capabilities, and compare the approach to other forms of adapting existing web content for mobile devices.

1.1 Evolution of mobile device standards

The wide array of mobile markup languages complicates mobile application development. The lack of vendor-independent standards prompted mobile device manufacturers to develop their own markup languages, such as Handheld Device Markup Language (HDML), Wireless Markup Language (WML) and Compact HTML (CHTML) [3]. While this allowed mobile devices to display some form of markup language, it fractured web development for mobile devices.

In an effort to create a standard markup for wireless devices, manufacturers merged the feature set of XHTML Basic [4] and wireless specific markup

languages, creating two standards: XHTML Mobile Profile (XHTML-MP) and Wireless Markup Language (WML) 2 [5]. XHTML-MP contains a superset of XHTML Basic, but a subset of XHTML, and also supports CSS (a styling language standard by the W3C). On the other hand, WML2 contains XHTML MP and another set of features that were unique to mobile devices. Unfortunately, WML2 was not widely adopted, as it was regarded that mobile device features existed only for backward compatibility reasons.

2 Related work

2.1 Consensus – a similar project

The Consensus project [6], which aims to "enable programmers to develop browser based applications for a variety of different mobile devices by the usage of only one tool," mirrors what we want to achieve, albeit having a larger scope. However, while the Consensus project was able to produce respectable results, the actual implementation, which involves an array of open source products and a custom markup language called Renderer Independent Markup Language (RIML), is too complex to build upon. In addition, adopting RIML requires web developers to rewrite existing content in a custom syntax, an expensive and risky task.

The Consensus project inherently supports pagination through their RIML markup language, in which pagination attributes are explicitly defined. Central to authoring RIML is the concept of containers; containers are used to layout the presentation of a page, and can be paginating or non-paginating. The actual content is located in frames. Frames further consist of sections, which define the logical grouping of content. In addition, actual pagination links are generated according to the navigation element, which specify the scope of the pagination. Together with separation of page structure and page content, RIML offers a flexible solution for serving multi-structured content to a wide array of devices via a single authored document, at the cost of increased complexity and whole new content-authoring model.

Comparing the Consensus model to our approach, we see that Consensus is much more flexible in terms display options for the content. Because traditional HTML authoring does not have a clear separation of document content and structure, it is not possible to define multiple page structures in the same source page. However, HTML meta-data capabilities allow web developers to achieve similar content grouping functionality to Consensus. Moreover, we are proposing a much more evolutionary approach that takes advantage of the inherent meta-data capabilities of XHTML-MP.

2.2 Other approaches

Other researchers and domain experts have produced work relating to the problem of adapting content for mobile devices, and have influenced the direction of this paper.

Buyukkokten et al. [7] proposed a novel approach to formatting content: to shrink or expand identified content nodes, or Semantic Text Units (STU). These STUs are identified using several heuristics, and would summarize the content by the first line using keywords. In this approach, which Buyukkokten calls "Accordian Summarization", the user is able to shrink and expand content based on their relevancy. While it is not without merit, "Accordian Summarization" does not take advantage of the graphic displaying capabilities that most mobile devices today have, and does not take advantage of the XHTML-MP format.

McKeown et al. [8] proposes an algorithm that, in general, detects the largest text block and sets it as content. This approach is not consistent with what we have in mind, and likely provides poor results for complex pages.

Rahman et al. [9] advocates structural and contextual analysis, as well as summarization. The emphasis is on segmenting content based on HTML "zones", and to apply attribute based analysis. The relationship between zones would then be analyzed, and the content reorganized. Although Rahman et al. give useful tips to parsing and formatting HTML content, the work mainly serves as a guideline, and the approach has not been implemented.

A study by Kaasinen et al. [10] details two different reports: an attempt to build a web application from the ground up for a wireless mobile device, and the transformation of existing web pages to a wireless device content format. What is particularly interesting is that the final format for both experiments in the report use (WML) 1.0, an older standard for authoring wireless content. While the report does not give many details about the implementation of their experiments, it raises many issues regarding the usability of transforming pages into WML.

3 Our approach

We approach the problem of transforming existing web pages into wireless content by dividing it into three separate problems: conversion of unstructured or invalid web pages into valid pages, transformation of valid pages into a markup suitable for mobile devices, and finally, formatting of the resulting documents such that they are suitable for display on wireless devices. The bulk of this paper focuses on the transformation and formatting issues of the framework, though web page validation and correction is briefly explored.

3.1 Correction of invalid web pages

The proliferation of invalid web pages can be partially attributed to early implementations of HTML, which were very much vendor dependent; web pages produced different rendering results depending on the web browser they were running on, causing developers and designers to create browser specific web pages. To exacerbate the problem, versions of popular WYSIWIG (What-You-See-Is-What-You-Get) web design tools such as Macromedia Dreamweaver [11] and Microsoft FrontPage [12] often produce proprietary or invalid code. Despite the emergence of standards proposed by the W3C such as Extensible Markup

Language (XML) [13], Extensible Hypertext Markup Language (XHTML) [14], and Cascading Style Sheets (CSS) [15], a large proportion of web pages remain invalid today. Indeed, the Web Standards Project (WASP [16]) was established to promote core web standards and browser compliance as a direct response to this problem.

Given the cause of most invalid web pages, we explored existing tools that could be leveraged to facilitate the automated transformation of invalid and unstructured pages into valid XHTML. Our initial effort to extend an open-source utility called TIDY [17] produced disappointing results. While we did not pursue the issue further, the increased awareness of web standards and the adoption of standards compliant browsers lead us to believe that the proportion of standards compliant XHTML documents will increase.

3.2 Transformation of valid web pages into a wireless markup

While our initial intent was to convert to the baseline XHTML Basic standard, we decided that the advanced XHTML MP was a more suitable target format. In addition to being supported by the major mobile browser vendors, XHTML MP contains a richer feature set, and most importantly, supports CSS for the separation of web page content and web page styling. (The CSS support not only takes advantage of the graphical capabilities of modern mobile devices, but also allows developers to hide content unsuitable for mobile devices, as explained in the section on formatting.)

Given that the input page is an XML based document (XHTML Transitional in our case), the task of conversion to XHTML-MP is straightforward; we decided to implement it via XSLT [18], an XML transformation standard. XSL is a language and technology for transforming XML based documents; by matching certain nodes or attributes in an XML document, developers are able to update them with information specified in the XSLT style sheet.

The W3C provides a XSLT style sheet that converts an XHTML document to XHTML Basic, which we use as a basis for creating a style sheet that converts from XHTML to XHTML Mobile Profile. Unfortunately, the style sheet that the W3C provided did not always generate correct XHTML Basic code. For example, one stringent requirement for XHTML Basic and XHTML MP is that nested tables, a mechanism often used to style web pages by web developers, are not allowed; the W3C style sheet does not enforce this rule. We resolve this issue by converting tables into DIV blocks and positioning the blocks using CSS. With the issue of nested tables resolved, it us only a matter of removing some filters in the original XSLT style sheet to make the style sheet suitable for the transformation into XHTML Mobile Profile, which, as previously mentioned, is a strict superset of XHTML Basic.

3.3 Formatting content for mobile devices

3.3.1 Deck-of-cards

The shared syntax and overlapping feature set between XHTML and XHTML-MP allow web page developers to reuse their existing skill set. However, as

previously mentioned, simply producing a valid XHTML-MP document will not suffice for mobile browsers. In particular, the smaller screen size and limited navigation capabilities of mobile devices suggest that content must be formatted in a wholly different manner.

To produce a viable formatting algorithm, a suitable navigation approach must be chosen. Two main navigation approaches are considered for the purposes of this project: a two-level image thumbnail approach, and a deck-of-cards approach.

The image thumbnail approach provides a navigation solution that retains the original structure and formatting of a page. In general, users are first shown an image thumbnail, which serves as an index to the various sections of the page. Upon clicking on the appropriate section of the page thumbnail, the browser "zooms in" and the selected section is enlarged.

Chen et al. [19] proposed an implementation using a multi level approach to detect and format content: Under the approach, high-level blocks are detected using the Document Object Model (DOM) [20]. Low-level blocks are detected using heuristics and are based on several pattern recognition rules. While this method allows for a straightforward transformation of existing pages while maintaining the structure of the source page, it should be note that the approach is aimed towards PDA-class mobile devices, which have touch screen or mouse-like input.

On the other hand, the deck-of-cards approach originates from a navigation metaphor used in WML. In WML, rather than grouping content by pages, related content can be placed in the same "deck" of cards, with each card representing a unit of content. For example, a singe form may span multiple cards, as illustrated in Figure 1.

By spreading the form across multiple cards, usability is increased by reducing vertical scrolling [21]. In addition, this form of navigation may help reduce bandwidth requirements: by downloading whole WML decks rather than individual cards at a time, the mobile browser may reduce round trip time to the server and thereby improve the user navigation experience between cards. An implementation of the deck-of-cards metaphor then, would paginate sources pages into multiple cards while providing a means to navigate between the cards.

Ultimately, we decided to follow the deck-of-cards metaphor based on the following rationale: while rendering capability of mobile devices are likely to increase, most of them will still be constrained by their limited screen sizes. The image thumbnail approach is suitable for devices with slightly larger screens, such as personal digital assistants, but not other wireless handheld devices such as cell phones, which are becoming even more prevalent than PDAs. Since we do not want to discount the most popular type of web enabled mobile device (mobile phones), the deck-of-cards approach is more appropriate.

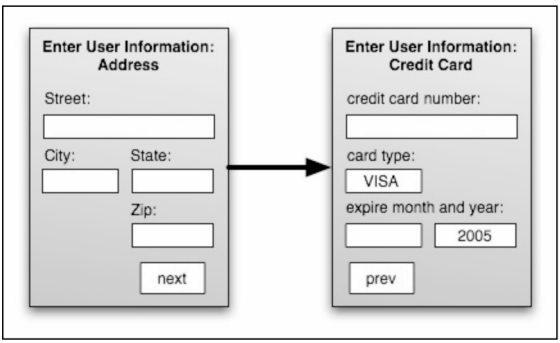

Figure 1: Example form spanning 2 cards.

3.4 Pagination

Pagination, in this case defined as the process of breaking logical blocks of content into separate pages, plays a key role in any solution for serving web content to multiple devices. Our implementation of the deck-of-cards metaphor is based on 2 modules: a pagination module and an optional Multipart module.

The pagination module is responsible for processing source XHTML-MP documents into a paginated form. Using the open source JDOM [22] package, the module manipulates a tree representation of the source document. To facilitate pagination, source documents must be extended to contain meta-data that define logical groupings of units that cannot be separated, called "sections". Sections are denoted in the class attribute an HTML tag. In addition, the id attribute of the containing tag of the section must be denoted in a HTML meta tag. Figure 2 shows a sample document in which containers and sections are denoted.

```
<!DOCTYPE html PUBLIC "-http://WAPFORUMhttp://DTD XHTML Mobile 1.0http://EN"
"http:http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
<html>
    <head>
    <meta name="container" content="somecontainer" />
    </head>
    <body>
    <div id="somecontainer">
        <div class="block $section">
        ...
        </div>
        <div class="block $section">
        ...
        </div>
    </div>
    </body>
</html>
```

Figure 2: Denoting sections and containers.

Figure 3: Source page (borders show paginated section).

Figure 4: Paginated page (partial listing).

Pagination occurs solely on the denoted containers; depending on the device screen size, each paginated page contains a multiple sections of content. The content besides container tree nodes and their sections are copied to each paginated page as part of the pagination process, effectively serving as a template.

Given such a logical grouping, a proxy application that is aware of client device specifications may choose the most suitable pagination. Although our

current implementation simply paginates one section per page, a more realistic algorithm may entail server side user agent sniffing through HTTP headers. Alternatively, the client may specify his device profile in a web form prior to accessing the actual content. Navigation links between paginated pages are automatically generated for each container; Figures 3 and 4 show the resulting pagination of a page (only a single paginated page is shown to reduce clutter).

3.5 Bandwidth conservation

As explained in Section 3.3.1, the WML implementation downloads whole decks-of-cards at a time, thereby conserving download bandwidth. While this is not possible with the HTTP protocol, a similar effect can be achieved by taking advantage of the Multipart MIME support of certain mobile browsers [23]. We implement this in the Multipart module, which aggregates the resulting pages (or cards in the deck-of-cards metaphor) and produces a single Multipart MIME document, thereby reducing download latency. However, because forms may only span one page in HTML 1.0, it is not possible to incrementally update a form on separate cards (as is possible with WML, see Figure 3). This is likely to change as the next generation of XHTML, XHTML 2.0 [24], becomes adopted. A preliminary draft of XHTML 2.0 indicates support for XFORMS [25], which separates the presentation and purpose of a form, thereby allowing forms that span pages.

3.6 Content filtering

It is very common for authors to filter the content of a source page for a target mobile device. For example, ads or large graphics may be removed to reduce clutter. While it is possible to add the option to remove flagged content as part of the pagination process, we take advantage of existing CSS capabilities to achieve the same effect. Indeed, by detecting the handheld media type and loading the appropriate style sheet containing a “display:none” rule, undesired content may be filtered out.

4 Conclusion

As the market for internet-enabled hand held devices continues to grow, the importance of adapting existing content for mobile browsers will increase as well. Indeed, the variation among mobile devices has led to various methods for the presentation of data. By leveraging existing standards such as HTML Mobile Profile and CSS, our framework takes an evolutionary approach for adapting content for mobile web pages, and presents a possible solution for organizations wishing to adapt their existing content for mobile devices in a gradual manner.

References

[1] Dubost, K. My Website is Standard. And Yours? http://www.w3.org/QA/2002/04/Web-Quality, 2005

[2] Parnas, D., How to cope with incorrect HTML. University of Bergen, pp 84, 2001
[3] An Overview of Mobile Versions of XHTML. Little Springs Design Inc., http://www.littlespringsdesign.com/design/xhtmlinfo
[4] XHTML Basic, http://www.w3.org/TR/xhtml-basic
[5] Open Mobile Alliance, http://www.openmobilealliance.org
[6] Consensus Home Page, http://www.consensus-online.org
[7] Buyukkokten, O. & Garcia-Molina, H. & Paepcke, A,"Accordion Summarization for End-Game Browsing on PDAs and Cellular Phones", ACM Press, http://portal.acm.org/citation.cfm?id=365102
[8] McKeown, K. R. et al., "Columbia Multi-Document Summarization: Approach and Evaluation", Document Understanding Conference 2004, http://www-nlpir.nist.gov/projects/duc/pubs.html, 2004
[9] Rahman, A. F. R. et al., "Content Extraction from HTML Documents", http://www.csc.liv.ac.uk/~wda2001/Papers/11_rahman_wda2001.pdf, 2001
[10] Kaasinen, E. "Two approaches to bringing Internet services to WAP devices", WWW1999 Conference, http://www9.org/w9cdrom/228/228.html, 1999
[11] Dreamweaver product page. Macromedia Inc., http://www.macromedia.com/software/dreamweaver/, 2005
[12] Frontpage 2003 product page. Microsoft Inc., http://r.office.microsoft.com/r/rlidAppFolder?clid=1033&p1=frontpage, 2005
[13] XML, http://www.w3c.org/XML/
[14] XHTML 1.0, http://www.w3.org/TR/xhtml1/
[15] CSS, http://www.w3.org/Style/CSS/
[16] WASP: Fighting for standards. Web Standards Project, http://www.webstandards.org/about/
[17] HTML Tidy, http://tidy.sourceforge.net/
[18] XSLT, http://www.w3.org/TR/xslt
[19] Chen ,Y. & Ma, W. & Zhang, H., "Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices", WWW2003 Conference, 2003
[20] Document Object Model, World Wide Web Consortium, http://www.w3.org/DOM/
[21] Kaasinen, E. "Two approaches to bringing Internet services to WAP devices", WWW1999 Conference, http://www9.org/w9cdrom/228/228.html, 1999
[22] JDOM, http://www.jdom.org/
[23] Converting WML to XHTML, Openwave Inc., http://developer.openwave.com/dvl/support/documentation/technical_notes/wml2xhtml.htm
[24] XHTML 2.0, http://www.w3.org/TR/xhtml2/
[25] XForms, http://www.w3.org/MarkUp/Forms/