An Efficient and Systematic Method to Generate XSLT Stylesheets for Different Wireless Pervasive Devices

Thomas Kwok, Thao Nguyen, Linh Lam and Kakan Roy
IBM Research Division,
Thomas J. Watson Research Center, Hawthorne, NY 10532
1-914-784-7650
kwok@us.ibm.com

ABSTRACT

It is a tedious and cumbersome process to update directly a WML document for the wireless Web because its content composes of both data and presentation. Thus, XML is used to handle the data while its XSLT stylesheet is used to extract and format the data for presentation. However, different stylesheets have to be used for different devices. An efficient and systematic method based on the idea of generating two separate sets of rules corresponding to content extracting and formatting parts of the stylesheet is described in this paper. The data extraction part is constructed from content rules while the formatting part is constructed from presentation rules. They are then combined together to form a stylesheet by an XSLT generator. A large number of stylesheets corresponding to different devices and a number of standard DTD documents or XML schemas can be generated in this way and stored in the pool during application setup stage. They will be individually selected from the pool by an XSLT engine to produce different WML documents for different devices during run time.

Categories & Subject Descriptors: D.1.1 [**Programming Techniques**]: Applicative Programming.

General Terms: Algorithms, Documentation.

Keywords: XSLT, WML, XML, PDA, pervasive devices.

1. INTRODUCTION

There are some limitations of wireless pervasive devices in displaying published content from the wireless Web [1] such as the small screens available on most cellular phones and PDAs. Wireless Application Protocol (WAP) [2] has defined a new format, the Wireless Markup Language (WML) for content delivery to wireless pervasive devices. However, it is a tedious process to update a WML document because the content composes of both data and presentation format. Thus, Extensible Markup Language (XML) [3] is used to handle only the content data. XML documents are validated by a Document Type Declaration (DTD) document or an XML schema. XML has its own style language called Extensible Style Language (XSL). It provides a standard way of extracting what information in an XML document should be included in the presentation, and expressing how this information should be presented. XSL consists of two parts, a transformation language and a formatting language. The transformation language is used to transform documents into different forms, while the formatting language is used to actually format and style documents in various ways. This transformation language is called Extensible Style Language Transformations (XSLT). An XSLT stylesheet is used to specify the exact format of the presentation. It can be read by a Java servlet in the server, a browser or any XSLT engines. An example of a XSLT engine is the IBM Websphere Transcoding Publisher (WTP) [4].

However, wireless pervasive devices including cellular phones and PDAs come with different form factors. They vary greatly in display size, keypads, supported functions, software keys and capabilities. Moreover, the differences among their micro-browsers are large enough to have their GUI modified. Thus, it is very difficult to construct a single XSLT stylesheet to produce a single WML, which can then be used for different WML enable wireless devices and their micro-browsers. Currently, different XSLT stylesheets have to be manually constructed for different PDAs or cell phones. In the setup of a new wireless application, different WMLs have to be generated from these different XSLT stylesheets by the XSLT engine. During run time, the servlets, JSPs or ASPs can identify the device client by detecting two HTTP header values: User-Agent and Accept. Then, different XSLT stylesheets are chosen for the XSLT engine to produce different WML documents for different devices. As a result, the application developer has to construct manually quite a large number of XSLT stylesheets just for one wireless application. For example, there are (L x M x N) XSLT stylesheets required just for an application if there are L wireless pervasive devices, M XML documents for the application and N average number of WML desks for each XML document.

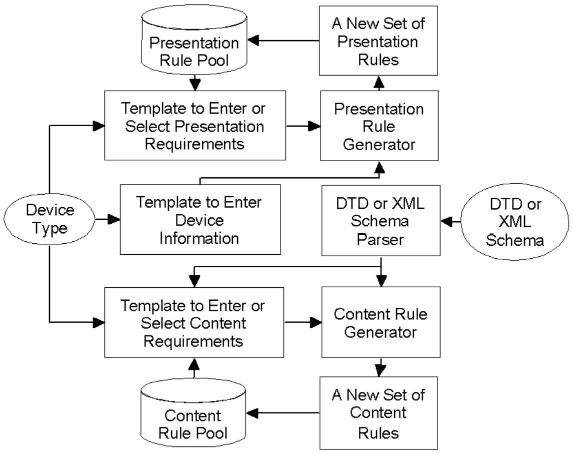
Presentation Rule Pool
A New Set of Prsentation Rules
Template to Enter or Select Presentation Requirements
Presentation Rule Generator
Device Type
Template to Enter Device Information
DTD or XML Schema Parser
DTD or XML Schema
Template to Enter or Select Content Requirements
Content Rule Generator
Content Rule Pool
A New Set of Content Rules

Figure 1. Creation of Content and Presentation Rules Pools.

2. GENERATE XSLT STYLESHEETS

As shown in Figure 1, an XPath extraction layer in a DTD or XML Schema extracts all different XPath from a DTD document. Then, the content rule generator constructs a new set of content rules based on all the tags under each XPath. A template is provided for the user to enter device information such as screen

WWW 2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

size, number of lines, number of characters per line, and supported software keys or functions for the presentation rule generator to construct a new set of presentation rules. These two new sets of rules are stored in their corresponding pools. Alternatively, a template can be provided for the user to enter directly or select a set of presentation or content requirements for the presentation and content rule generators to generate a new set of presentation and content rules.

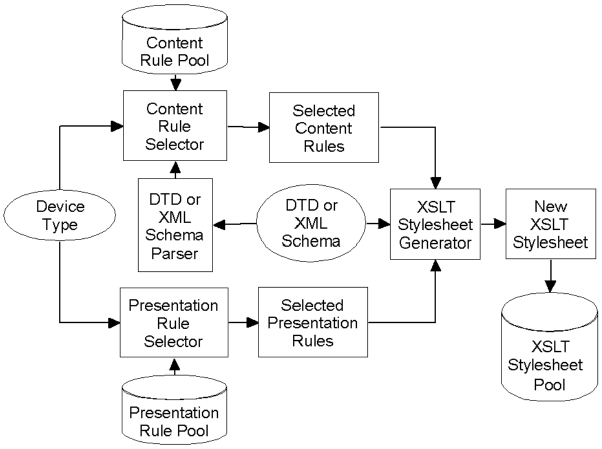

Figure 2. Generate XSLT Stylesheets.

In an application setup stage, the content rule selector is to select a set of content rules based on the device information and the DTD document or XML schema from the pool as shown in Figure 2. A set of presentation rules is also selected based only on the device information. Then, the XSLT stylesheet generator generates a stylesheet by combining these two sets of presentation and content rules. The content selection or extraction part is constructed from content rules while the presentation style part is constructed from presentation rules. The XSLT stylesheet generated is then stored in the stylesheet pool for run time use.

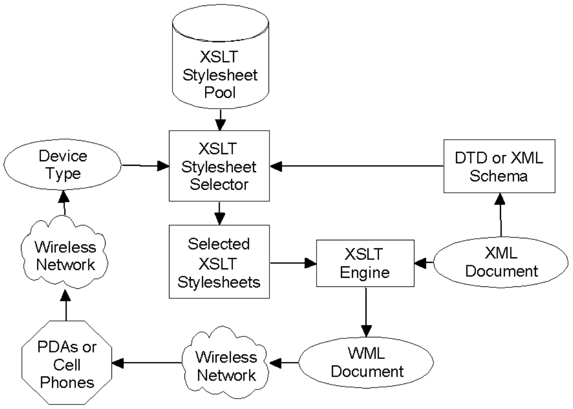

Figure 3. Produce XML Documents.

3. PRODUCE WML DOCUMENTS

During run time, the device information of a wireless pervasive device defined from WAP header is passed to the XSLT stylesheet selector as shown in Figure 3. Then, the selector selects a stylesheet with respect to the device information and the DTD document or XML schema from the pool as shown in Figure 3. With input of this stylesheet and a XML document, a WML document of an appropriate content and presentation style is generated for this device by an XSLT engine. Alternatively, an XSLT stylesheet can be dynamically generated each time when needed by the XSLT engine. In this case, a content rules generator is required to generate a set of content rules from either the input of a content only XSLT stylesheet or the input of a XML document with the help of a XML parser. Moreover, our method can handle output documents of other formats, such as HDML, i-Mode CHTML, Palm OS HTML and Voice XML for different wireless pervasive devices in a similar way.

4. DISCUSSION AND CONCLUSION

When trying to convert a wired application to run on the wireless device, the major effort is to redesign and rewrite the presentation part of the application so that the content would still show properly on the wireless device. The presentation part of an application could be handled by different stylesheets to fine tune the content to fit different types of devices. This process could become very lengthy and cumbersome depending on how many device types that the application intends to support. Thus, developers could use a GUI tool to choose what and how they would want to show the content for the application that is best for a specific device type. Then, our method would automatically generate the stylesheet to support this device type. This automated process could significantly shorten the development cycle.

Moreover, wireless pervasive devices tend to support their own proprietary formats. For developers to port an application to run on all different types of devices, typically they need to understand the syntax and rules of these different proprietary formats before they could even start the process of writing stylesheets for these devices. Since some of these formats could be quite tedious, it may take a lot of time and efforts to have all stylesheets ready before the application deployed for different types of devices. The method described in this paper allows developers to generate different formats without having to know the tedious details of each format. Hence, our method reduces the developers' overhead of learning different formats and makes porting wired applications to wireless devices a less time consuming process.

5. ACKNOWLEDGEMENTS

The authors would like to acknowledge the review of the manuscripts, suggestions for improvement and support from Daniel Dias and Lorraine Herger.

6. REFERENCES

[1] Alesso, H. Peter., and Smith, Craig F., ***The Intelligent Wireless Web***, Addison-Wesley, New York, 2002.

[2] Dornan, Andy., ***The Essential Guide to Wireless Communications Applications, From Cellular Systems to WAP and M-Commerce***, Prentice Hall, New York, 2000.

[3] Goldfarb, Charles F., and Prescod, Paul., ***XML Handbook***, Prentice Hall, New York, 2001.

[4] IBM Websphere Transcoding Publisher Version 1.1: Extending Web Applications to the Pervasive World, SG24-5965-00, IBM Redbook, 2000.