THE DESIGN AND IMPLEMENTATION OF A SEMANTICS-BASED WEB CONTENT ADAPTATION FRAMEWORK FOR MOBILE DEVICES

Chichang Jou and Cheng-Ze Ho
Department of Information Management, Tamkang University, Tamsui, Taipei, Taiwan

Keywords: Content adaptation, semantic web, CC/PP, Jena.

Abstract: With the rapid development of wireless communication technology, many users are accessing the internet from mobile appliances, such as notebooks, PDAs, and cellular phones. These devices are miscellaneously limited in computing resources, like CPU speed, memory, temporary storage, power supply, installed software, and communication bandwidth. The web accesses in these mobile devices thus encounter distorted user interface, broken images, slow responses, etc. To overcome this problem, we design and implement a semantics-based web content adaptation framework to provide automatically adapted contents to miscellaneous mobile devices. RDF Semantics of client device CC/PP configurations and web page tag structures are extracted to determine the proper parameters for the format and layout of web contents. Heuristic transcoding rules in the Jena Inference System are then applied to transform the web contents for each particular device. The functionality of this framework is illustrated by its capability of adjusting the layout of the main page of a shopping portal and of adjusting the parameters of images in the page.

1 INTRODUCTION

With the rapid development of wireless communication technology, many users are accessing the internet from mobile appliances, such as notebooks, PDAs, and cellular phones. Many emerging computation paradigms, such as pervasive computing, have been proposed to embrace this blooming portable computation trend. However, mobile devices have various hardware limitations, such as CPU speed, power, memory, and image resolutions. They are also restricted in software support, such as operating system, installed programs, real-time processing capability, and rendering functionality. These *ad hoc* limitations have become barriers in human-computer interaction. Especially, current internet contents, such as web pages and images, are mainly in the HTML format designed for desktop computers. Without any modification, it is hard to render them properly in most mobile devices.

In this paper, we propose a semantics-based web content adaptation framework to overcome the presentation inconsistency among mobile devices and desktop computers. In this framework, web pages and image files are automatically transcoded according to semantics extracted from CC/PP (Composite Capability/Preference Profiles) (Klyne et al., 2004) client device configuration and web contents. These semantics properties will be stored inside the Jena Inference System (Carroll, et al., 2004) as knowledge facts to infer proper transcoding parameters for each particular device. The adapted web pages with proper layout modification and images with proper rendering parameters will be constructed and delivered. Thus, this technology is suitable for resource-limited devices to balance information loss and information availability.

For the rest of this paper, we will review related work in Section 2. In Section 3, we describe the system architecture of this framework. Semantics extraction and knowledge base construction for the Jena Inference System are discussed in Sections 4 and 5. Section 6 demonstrates the content adaptation function with respect to the generated transcoding parameters. Section 7 explains the system implementation and illustrates through three mobile devices examples of rewriting one web page, and of adjusting the width and height parameters of images in that page. Section 8 concludes this paper.

Jou C. and Ho C. (2007).
THE DESIGN AND IMPLEMENTATION OF A SEMANTICS-BASED WEB CONTENT ADAPTATION FRAMEWORK FOR MOBILE DEVICES.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Web Interfaces and Applications*, pages 106-113
DOI: 10.5220/0001291801060113

2 RELATED WORK

Many researchers have tackled the incompatibility issue of mobile device in accessing web pages. Bickmore and Girgensohn (1999) designed a "Digestor System" which is capable of automatic filtering and re-authoring so that WAP-enabled cellular phones could read HTML contents. Their basic idea is to extract plain texts in the HTML document by discarding all formatting elements and unnecessary information. The result is then divided into a navigation page and several plain text sub-pages. They also utilize transcoding cache to diminish the run-time overhead.

Ardon et al. (2003) prototyped a proxy-based web transcoding system based on network access control, user preferences, and displaying capability of equipments. Since all transcoding procedures were finished in the content provider's server, this server-centric framework avoided potential copyright problems. Lum and Lau (2002) built a quality-of-service oriented decision engine for content adaptation. They designed flows for content negotiation and processing for multimedia contents. Hua et al. (2006) integrated content adaptation algorithm and content caching strategy for serving dynamic web content in a mobile computing environment. They constructed a testbed to investigate the effectiveness of their design in improving web content readability on small displays, decreasing mobile browsing latency, and reducing wireless bandwidth consumption.

Buyukkokten et al. (2001) used an "accordion summarization" transcoding strategy where an HTML page could be expanded or shrunk like an accordion. The HTML page is restructured as a tree according to the semantic relationships among its textual sections. All textual sections are split into several Semantic Textual Units, which are automatically summarized. Users can check each summary to expand the node for detailed information. However, this framework only works in the browser they designed for digital libraries.

Hwang et al. (2003) also treated web page layout as a tree according to its tag hierarchy. They defined a grouping function to transform such tree into sub-trees, and introduced a filtering mechanism to modify the sub-trees for adequate display in the target device. They analyzed specific web page layout structure and re-authored, according to heuristics, web pages for several mobile devices. Each of their transcoding method could handle only specified layout structures of web pages and did not consider mobile device characteristics.

Huang and Sundaresan (2000) tried the semantics approach in transcoding web pages to improve web accessibility for users. Their system was designed to improve the interface of e-business transactions and to extend interoperable web forms to mobile devices. They used XML/DTD to specify the semantic and grammatical relationship among web contents, so that web forms could achieve consistency, simplicity and adaptability. The advantage of this system is its ability to provide concept-oriented content adaptation, but it is hard to extend.

DELI (Butler, 2002), an HP Semantic Lab project, adopted simple negotiation algorithms for rewriting web pages based on context information, like user preference and device capabilities. Glover and Davies (2005) used heuristic algorithms to find proper pre-defined web page templates according to device attributes. Their focus is in applying XML/XSLT styles to database contents retrieved in dynamic web pages.

In our framework, we follow the guideline of the "Device Independent" (Bickmore et al., 1997) principle and utilize the device contexts described in CC/PP.

3 SYSTEM ARCHITECTURE

In this section, we explain the system architecture and the data flows of our framework in Figure 1. The adaptation mechanism of our framework, called Content Adaptation Proxy Server, resides behind web servers. The Proxy Listener is responsible for receiving HTTP requests (message 1) from miscellaneous mobile devices and for dispatching these requests to the Content Fetcher. The client's device information as well as user's personal preferences will be embedded inside these requests through CC/PP diff, which is a modified version of predefined CC/PP profile from the hardware manufacturers.

CC/PP is a two-layered user preferences and device capabilities description based on XML/RDF (Manola and Miller, 2004). CC/PP consists of the following three categories: hardware platform, software platform, and browser user agent. Figure 2 shows example detailed attributes in the XML/RDF format of one mobile device. (Klyne et al., 2004) In our framework, all predefined CC/PP profiles are stored within a profile directory. A CC/PP diff is manually configured by the user, and is normally used to reflect the user preferences. It is dynamic and can be further modified in later sessions. Many protocols have been proposed to enhance HTTP 1.1

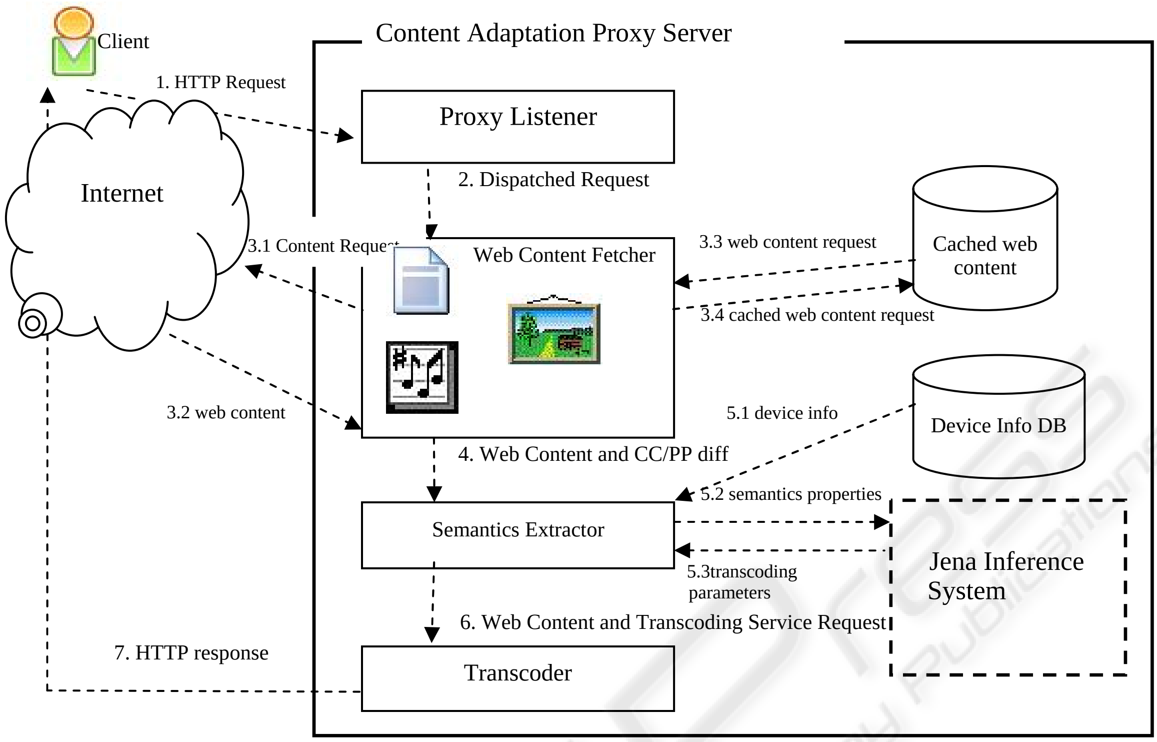

Figure 1: System Architecture of Content Adaptation Proxy Server.

protocol to include CC/PP profile diff. Two of such protocols are CC/PP-ex (Ohto and Hjelm, 1999) and W-HTTP (OMA, 2001). We adopt CC/PP-ex in this framework.

```
[ex:MyProfile]
 |
 +--ccpp:component-->[ex:TerminalHardware]
 |                    |
 |                    +--rdf:type----> [ex:HardwarePlatform]
 |                    +--ex:displayWidth--> "320"
 |                    +--ex:displayHeight--> "200"
 |
 +--ccpp:component-->[ex:TerminalSoftware]
 |                    |
 |                    +--rdf:type----> [ex:SoftwarePlatform]
 |                    +--ex:name-----> "EPOC"
 |                    +--ex:version--> "2.0"
 |                    +--ex:vendor---> "Symbian"
 |
 +--ccpp:component-->[ex:TerminalBrowser]
                      |
                      +--rdf:type----> [ex:BrowserUA]
                      +--ex:name-----> "Mozilla"
                      +--ex:version--> "5.0"
                      +--ex:vendor---> "Symbian"
                      +--ex:htmlVersionsSupported--> [ ]
                                                      |
                        ---------------------------
                        |
                        +--rdf:type---> [rdf:Bag]
                        +--rdf:_1-----> "3.2"
                        +--rdf:_2-----> "4.0"
```

Figure 2: Example CC/PP for a mobile device.

When Proxy Listener accepts a request from a client, it will spawn a working thread in the Web Content Fetcher to handle the request (message 2). Web Content Fetcher performs the standard task of proxy servers. If the requested web content is already in the cache, it will be fetched from Cached Web Content (messages 3.3 and 3.4). If not, then it will be fetched from the source through the internet (messages 3.1 and 3.2), and then be saved in the Cached Web Content. The working thread is also responsible for resolving the CC/PP profile diff.

The web content and CC/PP diff will then be sent (message 4) to the Semantics Extractor to acquire the implicit RDF semantic information within the CC/PP diff, HTML web pages and parameters of image files. RDF describes resource information in the internet by the three components: resource, property, and statement. A RDF statement is represented by a triple of (subject, predicate, object).

These semantics information will be sent (messages 5.2) to the Jena Inference System as basic facts. Jena Inference Engine will combine these facts with the knowledge base of CC/PP UAProf RDFS model (WAP, 2001), Transcoding Rules and Web Page Auxiliary Vocabulary to determine the proper transcoding parameters in the format of sequential RDF predicates for the requests.

The web content and transcoding parameters will then be passed (message 6) to the Transcoder.

Besides the layout rewriting mechanisms, the Transcoder is equipped with transcoding toolkits, such as ImageMagic (Still, 2005) for image resolution adjustment. The results of the Transcoder processing consist of web pages with modified layout as well as images with proper resolution and parameters for the requesting mobile device. They will be returned to the client (message 7) and displayed in its browser.

4 SEMANTICS EXTRACTOR

The Semantics Extractor is responsible of collecting information of the fetched web content by checking its file type, information of analyzing the structure of web pages, and information of constructing facts about the requested web contents and device characteristics.

Since most HTML pages are not well-formed, it is hard to extract semantics from them directly. This module first transforms HTML pages into the well-formed XHTML format through the Tidy toolkits (Raggett, 2000). The following two file types of the requested URL will first be handled in this framework: (1) XHTML files: Their metadata are about layouts of the document, possibly with hyperlinks to external textual or binary files. (2) Image files: These are binary files with adjustable parameters, like color depths and resolution. Currently, the system could handle JPEG, PNG, and GIF images. For files encoded in indestructible formats, like Java applets, since they could not be adjusted, the system directly forwards them to the Transcoder for delivery. To extend this web adaptation framework to new file types, we just need to specify the metadata about the new file type, and have to build the semantics extraction component and transcoding rules for web contents of the new file type.

For XHTML files, the semantic extractor module collects the following schema information: the identification of each XHTML element, and the layout of the XHTML page. We apply XHTML DOM (Document Object Model) (Le Hégaret et al., 2005) tree nodes scanning to extract node information and relationships among XHTML elements. We solve the element identification problem in an XHTML page by XPath (Clark and DeRose, 1999) so that each node in the DOM tree could be specified accurately.

Statistical or inferred semantics data for the following Web Page Auxiliary Vocabulary will be extracted for each XHTML DOM tree node:

1. **NumberOfWords**: This data indicates number of words in a paragraph. It is used to determine whether the paragraph corresponding to the XHTML node should be split.
2. **NumberOfImages**: This data indicates number of the <IMG> tags in a specific XHTML node. It is used to decide whether a tabular cell is an advertisement banner.
3. **AverageLink**: This is the quotient of the number of links and the number of words within a XHTML node. In web contents with useful information, this value tends to be very high, and all contents in the node should be preserved.
4. **Title**: For XHTML nodes with the <H1> or <H2> tag, or with texts surrounded by pairs of the <B></B> or <STRONG> </STRONG> tags and followed by
 immediately, the collected content is treated as a title. This could be used as the title of the sub-page corresponding to this node.
5. **Layout**: This information indicates whether the node is used for layout composition. For example, to determine whether a <TD> is a layout element or an actual tabular cell, we calculate the number of words for the element. If its number of words exceeds a specific threshold, we mark such a <TD> element as a layout element.

Consider the following simplified XHTML page:

```
<HTML><BODY><TABLE>
<TR><TD>Gentoo Linux is a totally new linux distribution.
</TD></TR>
</TABLE></BODY></HTML>
```

We can describe the <TD> tag in the above page with RDF, XPath and Web Page Auxiliary Vocabulary as follows:

```
<rdf:Description rdf:about="/HTML/BODY/TABLE/TD[1]">
  <rdf:type rdf:resource="html:ELEMENT_NODE" />
  <html:NumberOfWords>8</html:NumberOfWords>
  <html:IsLayout>false</html:IsLayout>
  <html:NumberOfImage>0</html:NumberOfImage>
  <html:NodeName>TD</html:NodeName>
  <html:ChildNodeNumber>0</html:ChildNodeNumber>
  <html:ParentNode rdf:parseType="Resource"
       rdf:resource="_:/HTML/BODY/TABLE/TR[1]"/>
</rdf:Description>
```

Semantics of device characteristics will be collected through CC/PP diff. The CC/PP semantics of device configurations are already RDF compatible. They could be sent to the Jena Inference System as facts. The total customized device description can be translated into a graph model within the Jena Inference System.

5 JENA INFERENCE SYSTEM

We use Jena (Carroll et al., 2004), a semantic web toolkit of "Device Independent" (Bickmore et al., 1997) ideal, to determine the transcoding parameters, which are represented as sequential RDF predicates. The Jena Inference System, displayed in Figure 3, has three main components. These components are utilized in our framework as follows:

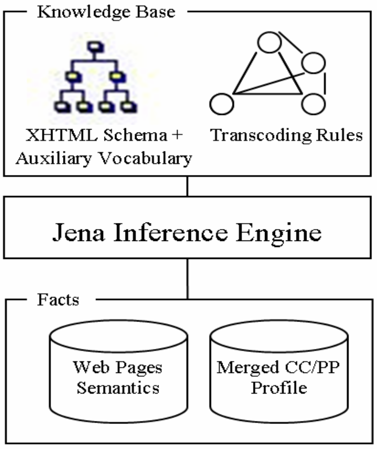

Figure 3: Jena Inference System.

Knowledge Base – It contains the acquired knowledge and rules in deciding the content adaptation parameters. XHTML schema is derived by mapping from XHTML XML schema to RDF/RDFS as one knowledge base. Transcoding Rules contain rules using web content ontology and device characteristics ontology for transcoding. Auxiliary Vocabulary for transcoding parameter decision are also described by RDF/RDFS and serialized into Jena knowledge base. All RDF knowledge is serialized in the XML format to provide more flexibility and interoperability in content adaptation.

Jena Inference Engine – This is the decision engine to inference and to generate transcoding parameters. We make use of the engine without any modification.

Facts – These are facts supported in the form of instantiated predicates. In our framework, they are the semantic data collected by the Semantic Extractor.

We apply heuristics to design the transcoding rules in the "IF…THEN…" format. If the precedent parts of a rule are all true, then the consequent part of the rule would be added as a statement of transcoding parameters into the knowledge base. We provide rules regarding device characteristics by defining restriction rules using first order predicate logic. Rules are categorized to back up each other. For example, if rules for HP 6530 PDA are not sufficient, then rules for PPC Pocket PC could be used. In the rules, names prefixed with '?' are variables. The following example rule is for image resizing:

```
[ScaleImageByWidth:
(system:Content content:Width ?image_width),
(system:HardwarePlatform ccpp:DisplayWidth ?display_width),
lessThan( ?display_width, ?image_width ) ->
(system:Content content:ScaleImageByWidth ?display_width)]
```

The rule is named *ScaleImageByWidth*. The three namespaces *"system"*, *"ccpp", and "Content"* point to the knowledge bases regarding the content adaptation proxy server, the standard UAProf Schema (WAP, 2001), and web content under transcoding, respectively. The above rule means that if the width of the device (?dispaly_width) is less than the width of the image (?image_width), then set the width of the image as the width of the device, and set the height of the image proportionally with respect to the adjustment ratio of the width.

6 TRANSCODER

The Transcoder is composed of several transcoding modules corresponding to file types of XHTML, JPEG, etc. It could be extended to handle other file types. According to the transcoding parameters, it performs content adjustment and filtering. For image files, currently the system not only transforms image files into the same format with different parameters, but also transforms image files into different formats.

The Transcoder dispatches the web contents according to their file type to the corresponding adaptation component. To perform the required content transcoding operation, it will query the inferred RDF model by the RDF query language RDQL to obtain the required transcoding parameters. The use of RDQL could prevent tight coupling of the transcoding components. An example RDQL query is demonstrated as follows:

```
SELECT ?predicate, ?object
WHERE ( system:Content, ?predicate, ?object)
USING system FOR http://www.im.abc.edu/~def/proxy.rdfs#
```

In RDQL, *USING* is to specify the name space. This query could obtain all RDF statements with subject *system:Content*, where *system* is the name space and *Content* represents web content currently under transcoding. Transcoding query results are represented as instantiated predicates. For example, if the transcoding predicate for an image file is *ScaleImageByWidth*, then the Transcoder would adjust the image width and height proportionally. After all RDQL query results are handled, the resulting web content would be returned to the client.

7 SYSTEM IMPLEMENTATION

We implement the content adaptation proxy server in the Fedora Linux 2.6.18 operating system by the Java Language J2SDK 1.5.0. We use the package org.w3c.dom as the class for handling XHTML DOM trees for web pages. The JPEG, PNG, and GIF image files are handled by the Java class java.awt.image.BufferedImage. We make use of the inheritance mechanism, so that the Content Fetcher interface only specifies one method:

public Object getContent( ).

Table 1: Transcoding rules for image files.

Transcoding Rule	Comment
ExtractColor: (system:Content content:Width ?image_color_depth), system:HardwarePlatform ccpp:ScreenColorDepth ?device_color_depth), essThan(?device_color_depth, ?image_color_depth) >(system:Content content:ReduceColorDepth ?device_color_depth), (system:ReduceColorDepth system:TranscodeType "text/jpeg")]	Modify image color depth to match the display capability of the device.
PngToJpeg: (system:Content content:Type “image/png”), system:SoftwarePlatform ccpp:CcppAccept ?Bag), noValue(?Bag ?li "image/png"), ?Bag ?li "image/jpeg") > (system:Content system:TransformTo "image/jpeg"), (system:TransformTo system:TranscodeType "text/jpeg")]	Transform PNG files to JPEG.
JpegToPlainText: system:Content content:Type “image/jpeg”), system:SoftwarePlatform ccpp:CcppAccept ?Bag), noValue(?Bag ?li "image/jpeg"), ?Bag ?li "text/plain") > (system:Content system:TransformTo "text/plain"), (system:TransformTo system:TranscodeType "text/jpeg")]	Transform JPEG to plain text.

Table 2: Transcoding rules for web pages.

Transcoding Rule	Comment
[ExtractTableContent: (?node content:NodeName "table"), (system:BrowserUA ccpp:TablesCapable "No") ->(system:Content system:ExtractTableContent ?node) , (system:ExtractTableContent system:TranscodeType "text/html")]	If the browser in the mobile device does not support table, then extract the content.
[FilterCSSScript: (?node content:NodeName "style"), (system:BrowserUA ccpp:StyleSheetCapable "No") -> (system:Content system:RemoveNode ?node)] *[FilterCSSScript: (?node content:NodeName "style"), (system:SoftwarePlatform ccpp:CcppAccept ?Bag), noValue(?Bag ?li "text/css") -> (system:Content system:RemoveNode ?node) , (system:RemoveNode system:TranscodeType "text/html")]*	If the browser in the mobile device does not support CSS, then filter the CSS content.
[FilterFlash: (?node content:NodeName "object"), (system:SoftwarePlatform ccpp:CcppAccept ?Bag), noValue(?Bag ?li "x-application/flash"), (?node content:InlineDocumentType “x-application/flash”) -> (system:Content system:RemoveNode ?node) , (system:RemoveNode system:TranscodeType "text/html")]	If the browser in the mobile device does not support Flash, then filter the Flash content.

The Semantics Extractor is implemented by a Java interface, a factory for semantic extraction, and one class for each supported file type. Some of the transcoding rules for images are listed in Table 1, and some of the transcoding rules for web pages are listed in Table 2.

To demonstrate the functionalities of this framework, we tested three client mobile devices: HP iPAQ hx2400, Symbian S80 Simulator, and Panasonic EB-X700. We would like to show the effect of the following two CC/PP parameters: supported file types and display size. The goal of the adaptation is to avoid the use of horizontal scroll bar, so as to increase the readability of transcoded pages and images. The related specifications and restrictions of these devices are listed in Table 3.

Table 3: Specifications and restrictions of mobile devices.

Device	HP iPAQ hx2400	Symbian S80	Panasonic EB-X700
Category	PDA	Smart Phone	Smart Phone
Operating System	Windows Mobile 5.0	Symbian Series80	Symbian Series60
Browser	IE Mobile	Built in	Built in
Supported file types	text/html text/css image/jpeg image/png image/gif	text/xhtml text/css image/jpeg image/png image/gif	text/chtml image/jpeg
Display size	480 x 320 (pixels)	220 x 640 (pixels)	176 x 148 (pixels)
Connection	Bluetooth	WLAN	GPRS

Figure 4 shows the upper part of the tested web page (http://www.amazon.com) in a Microsoft IE 6.0 browser in a desktop computer. The upper parts of the transcoded pages in the built-in browser for the three tested mobile devices are displayed by two screen shots in Figures 5 to 7. All resulting transcoded web pages satisfy the goal of avoiding the use of horizontal scroll bar by adjusting the page layout, image size, and image resolution. Unsupported CSS, Javascripts, flashes, div's and tables are filtered out.

8 CONCLUSIONS

We designed and implemented a flexible and robust infrastructure for web content adaptation using RDF semantics from CC/PP device characteristics, XHTML web pages, and JPEG, PNG, and GIF image files. Past researches in this area either did not take device characteristics into consideration, or were not a general purpose solution for miscellaneous mobile devices. We made use of the Jena Inference System to obtain the transcoding parameters through the fact and knowledge base built from the collected semantics.

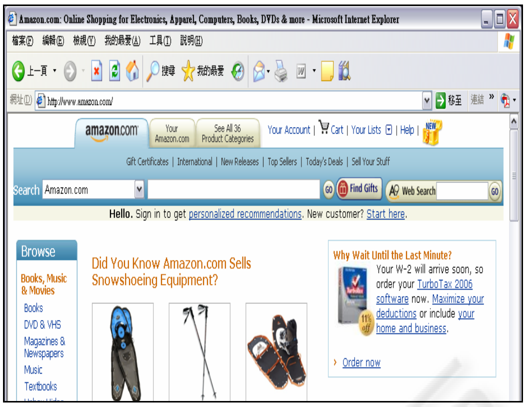

Figure 4: Test web page in a desktop computer.

Figure 5: Result of test page in HP iPAQ hx2400.

Figure 6: Result of test page in Symbian Series80.

Figure 7: Result of test page in Panasonic EB-X700.

In this framework, a single copy of the web pages could serve many different mobile devices. Previous tedious web page rewriting labour for mobile devices could be saved. Our framework could be easily extended to new file types by importing related semantics and transcoding modules.

We plan to incorporate support of style sheet and web form specifications, such as CSS and XForm, into this semantics-based content adaptation framework. By supporting these dynamic web pages, the increased user interactivity would accelerate user acceptance of pervasive computing.

ACKNOWLEDGEMENTS

This work was supported in part by Taiwan's National Science Council under Grant NSC 95-2221-E-032 -027.

REFERENCES

Ardon, S., Gunningberg, P., Landfeldt, B., Ismailov, Y., Portmann, M., Seneviratne, A., 2003. March: a distributed content adaptation architecture. In *International Journal of Communication Systems, Vol. 16(1), pp. 97-115.*

Bickmore, T., Girgensohn, A., 1999. Web page filtering and re-authoring for mobile users. In *The Computer Journal, Vol. 42(6), pp. 534-546.*

Bickmore, T., Girgensohn, A., Sullivan, J. W., 1997. Digestor: device-independent access to the World Wide Web. In *Proceeding of the 6th World Wide Web Conference WWW6, pp. 665-663.*

Butler, M., 2002. DELI: A Delivery context library for CC/PP and UAProf. In *External technical report, HP Semantic Lab.*

Buyukkokten, O., Garcia-Molina, H., Paepcke, A., 2001. Accordion summarization for end-game browsing on PDAs and cellular phones. In *Proceeding Of the SIGCHI Conference on Human Factors in Computing Systems, pp. 213-220.*

Carroll, J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K., 2004. Jena: implementing the semantic web recommendations. In *International 13th World Wide Web Conference, pp.74-83.*

Clark, J., DeRose, S., 1999. XML Path Language (XPath) v. 1.0. In *http://www.w3.org/TR/xpath, W3C.*

Glover, T., Davies, J., 2005. Integrating device independence and user profiles on the web. In *BT Technology Journal, Vol. 23(3), pp. 239-248.*

Hua, Z., Xie, X., Liu, H., Lu, H., Ma, W., 2006. Design and performance studies of an adaptive scheme for serving dynamic web content in a mobile computing environment. In *IEEE Transactions on Mobile Computing, Vol. 5(12), pp. 1650-1662.*

Huang, A.W., Sundaresan, N., 2000. A semantic transcoding system to adapt web services for users with disabilities. In *Proceeding of the 4th international ACM conference on Assistive technologies, pp. 156-163.*

Hwang, Y., Kim, J., Seo, E., 2003. Structure-aware web transcoding for mobile devices. In *IEEE Internet Computing, Vol. 7(5), pp 14-21.*

Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M., Tran, L., Editors, 2004. Composite Capability/Preference Profiles (CC/PP). In *http://www.w3.org/TR/CCPP-struct-vocab, W3C.*

Le Hégaret, P., Whitmer, R., Wood, L., Editors, 2005. Document Object Model (DOM). In *http://www.w3.org/DOM/, W3C.*

Lum, W.Y., Lau F.C.M., 2002. A context-aware decision engine for content adaptation. In *IEEE Pervasive Computing, Vol. 1(3), pp 41-49.*

Manola, F., Miller, E., Editors, 2004. RDF Primer. In *http://www.w3.org/TR/rdf-primer/, W3C.*

Ohto, H., Hjelm, J., 1999. CC/PP exchange protocol based on HTTP Extension Framework. In *http://www.w3.org/TR/NOTE-CCPPexchange, W3C.*

OMA, 2001. Wireless Profiled HTTP. In *WAP-229-HTTP-20011031-a of Open Mobile Alliance Ltd.*

Raggett, D., 2000. Clean up your web pages with HTML TIDY. In *http://tidy.sourceforge.net/docs/Overview.html, W3C.*

Still, M., 2005. *The Definitive Guide to ImageMagick.* Apress.

WAP, 2001. WAG UAProf. In *WAP-248-UAPROF-20011020-a of Wireless Application Protocol Forum Ltd.*