17

A Semantics-Based Mobile Web Content Transcoding Framework

Chichang Jou
Tamkang University
Taiwan

1. Introduction

With the rapid development of wireless communication technology, in addition to desktop computers, many users are accessing the internet from hand-held appliances, such as tablets, PDAs, and cellular phones. Many new computation paradigms, such as pervasive computing and ubiquitous computing, have been proposed to embrace this emerging portable computation framework. However, because of the difference in users' speed in adopting new technologies, hand-held devices in use have miscellaneous hardware limitations, such as CPU speed, power, memory, bandwidth, and image resolutions. They also have various restrictions 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. Most web contents, such as web pages and images, are mainly in the HTML format, which is designed for desktop computers. Without proper modification, most rendered web contents in hand-held devices encounter distorted or fragmented user interface, broken images, slow responses, etc. These ad hoc characteristics of hand-held devices have become a barrier in enhancing web availability.

In this chapter, we illustrate how a semantics-based content adaptation framework could be utilized to fill up the computational gap between mobile devices and desktop computers. The transcoding mechanism of our framework, called Content Adaptation Proxy Server [CAPS], resides behind web servers. In CAPS, web pages and image files are transcoded according to: (1) RDF (Resource Description Framework) (Manola et al., 2004) of web content; and (2) semantics extracted from the CC/PP (Composite Capability Preferences Profiles) (Klyne et al., 2004) client device configuration. These semantic properties will be stored and interpreted inside the Jena Inference System (Carroll et al., 2004) as knowledge facts to obtain proper transcoding parameters for each particular device. Then web pages with proper layout modification and images with proper rendering parameters for each particular device will be constructed and delivered. This technology will recreate contents suitable for resource-limited devices to balance information loss and information availability.

For the rest of this chapter, we will review related work in Section 2. In Section 3, we describe the design principle and system architecture of CAPS. Semantics extraction and knowledge base construction for the Jena Inference System are discussed in Sections 4 and 5.

Section 6 demonstrates web content transcoding process. Section 7 illustrates the implementation of CAPS and demonstrates transcoding examples on various mobile devices. Section 8 concludes this paper.

2. Related work

According how many web content types are handled, web content adaptation could be classified into: universal and specific web content adaptation. Universal web content adaptations are mostly proxy-based and could be applied to web pages and various multimedia types. Their functions include integrating various web content types for the rendering. Section 2.1 will cover proxy-based universal web content adaptation. Specific web content adaptation focuses on the algorithm design. The most frequently studied web content type is HTML files. Due to the complexity in analyzing HTML files, proper adaptation of HTML files in mobile devices is very difficult. Section 2.2 will cover works in using heuristics to adjust layouts of HTML documents to fit a particular mobile device. We will cover how the semantic web technology has been applied to web content adaptation in Section 2.3.

2.1 Proxy-based universal web content transcoding

Nagao et al. (2001) proposed constructing on the Web a system framework using XML and external annotations to Web documents. They proposed three approaches for annotating documents—linguistic, commentary, and multimedia. With annotated documents that computers can understand and process more easily, their framework allowed content to reach a wider audience with minimal overhead.

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.

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.

Sacramento et al. (2004) designed the Mobile Collaboration Architecture [MoCA], a middleware for developing and deploying context-aware collaborative applications for mobile users. It comprises client and server APIs, core services for monitoring and inferring the mobile devices' context, and an object-oriented framework for instantiating customized application proxies.

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.

Hsiao et al. (2008) proposed the architecture of versatile transcoding proxy (VTP). The VTP architecture can accept and execute the transcoding preference script provided by the client or the server to transform the corresponding data or protocol according to the user's specification. They adopted the concept of dynamic cache categories and proposed a new replacement algorithm for caches.

Nimmagadda et al. (2010) presented a content adaptation method for multimedia presentations constituting media files with different start times and durations. They performed adaptation based on preferences and temporal constraints specified by authors and generate an order of importance among media files. Their method can automatically generate layouts by computing the locations, start times, and durations of the media files.

2.2 Web page transcoding

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

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 was its ability to provide concept-oriented content adaptation, but it was difficult to be extended.

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 was restructured as a tree according to the semantic relationships among its textual sections. All textual sections were split into several Semantic Textual Units, which were automatically summarized. Users could check each summary to expand the node for detailed information. However, this framework only worked in the browser they designed for digital libraries.

Hwang et al. (2003) also treated web page layout as a tree according to the tag hierarchy. They defined a grouping function to transform such a 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.

2.3 Semantic web technologies in web content transcoding

In addition to Huang and Sundaresan (2000), several researchers have tried to incorporate semantic web technologies into web content transcoding. 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. Due to lack of implementation, its applications were restricted.

Hori et al. (2000) proposed an annotation-based system for Web content transcoding. They introduced a framework of external annotation, in which existing Web documents were associated with content adaptation hints as separate annotation files. This annotation-based transcoding system was then extended with particular focus on the authoring-time integration between a WYSIWYG annotation tool and a transcoding module.

Glover and Davies (2005) used heuristic algorithms to find proper pre-defined web page templates according to device attributes. Their focus was in applying XML/XSLT styles to database contents retrieved in dynamic web pages.

Hsu at el. (2009) proposed a hybrid transcoding approach to combine the traditional transcoding technologies based on ontology-based metadata to improve the rendering problem caused by heterogeneous devices. This heterogeneous markup document transcoding platform was then presented to serve as a transcoding service broker to facilitate interoperability between distributed heterogeneous transcoders.

3. Design principle and system architecture of CAPS

This section illustrates the design principle and system architecture of CAPS.

3.1 Design principle of CAPS

Butler (2001) describes the capabilities of mobile devices in the following categorizations: (1) output: screen, resolution, color, relative size of fonts, sound, etc. (2) input: touchscreen, mouse, keyboard, voice, joystick, etc. (3) processor (4) memory (5) multimedia objects: GIFs, JPGs, WAVs, MP3s, etc. (6) application language: native code, intermediate code. (7) browser language: content markup, client side scripting, applet, and styling. When a particular mobile device receives a web content that it could not handle, it should consider web content adaptation regarding the above aspects. However, from the users' point of view, a more important issue is whether the adapted content could be comprehended. To tackle the above two issues, CAPS follows the following guidelines of the device independence principle of W3C (Gimson et al., 2003):

- For some web content or application to be device independent, it should be possible for a user to obtain a functional user experience associated with its web page identifier via any access mechanism.
- A web page identifier that provides a functional user experience via one access mechanism should also provide a user experience of equivalent functionality via any other access mechanism.
- It should be possible for a user to provide or update any adaptation preferences as part of the delivery context.

3.2 System architecture of CAPS

In this section, we explain the system architecture and the data flows of Content Adaptation Proxy Server [CAPS] in Figure 1. Its adaptation mechanism 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 Web 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.

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 internet clouds (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 metadata of image files.

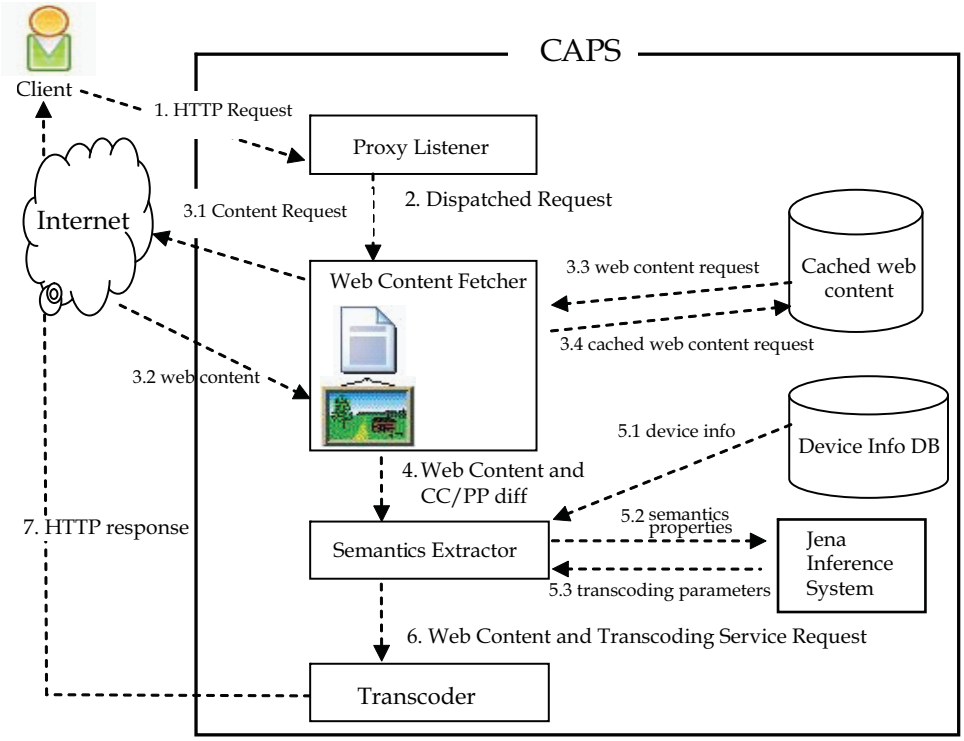


Fig. 1. System architecture of content adaptation proxy server

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 UAProfile RDFS model, 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 for image resolution adjustment. The results of the Transcoder processing consist of web pages with modified layout and images with proper resolution and parameters suitable for the requesting mobile device. They will be returned to the client (message 7) and displayed in its browser.

4. Semantics extractor of CAPS

This section illustrates details about semantics extracted from the CC/PP device configurations and the web content. Semantics of device characteristics will be collected through CC/PP diff. The CC/PP semantics of device configurations are RDF compatible already. 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, which will be described in Section 5.

4.1 Semantics extracted from device configurations

CC/PP is a two-layered user preferences and device capabilities description based on XML/RDF (Ohto & Hjelm, 1999). 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 (Klyne et al., 2004) for one mobile device.

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

Fig. 2. Example CC/PP for a mobile device

4.2 Semantics extracted from web content

Since most HTML pages are not well-formed, it is hard to extract semantics from them directly. The semantics extractor module first would transform HTML pages into the well-formed XHTML format through the JTidy[1] toolkits. The following two file types of the

[1]JTidy, http://jtidy.sourceforge.net

requested URL will be handled in CAPS: (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, CAPS could handle JPEG, PNG, and GIF images. For files encoded in indestructible formats, like Java applets, since they could not be adjusted, CAPS would directly forward them to the Transcoder for delivery to the client device.

To extend CAPS to new file types, we need to specify just the metadata about the new file type, and to build the semantics extraction component and transcoding rules for web contents of the new file types.

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 Document Object Model [DOM] (Le Hégaret et al., 2005) tree node scanning to extract node information and relationships among XHTML elements. We solve the element identification problem in an XHTML page by XPath (Clark & 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>
```

5. The Jena inference system of CAPS

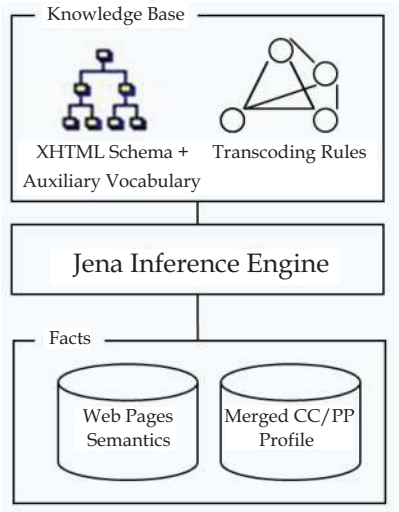


Fig. 3. Architecture of the Jena inference system

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

1. **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.
2. **Jena Inference Engine –** This is the decision engine to inference and to generate transcoding parameters. We make use of the engine without any modification.

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

The transcoding rules of CAPS are to link the web content description and device information in CC/PP, which are transmitted via CCPP diff. In other words, these rules would map the user's or device's requirements into parameter settings of web content. We follow the semantic network model (Hayes & McBride, 2004) defined by W3C, and define the following sets of facts to explain formal meaning of the transcoding rules:

1. **D**: CC/PP Facts transmitted from the client side. These are used to describe characteristics of devices or user preferences.
2. $\mathbf{C_t}$: All web content semantics for MIME type t obtained by "semantics extractor".
3. **L**: All constraints and limitations of web content. For example, image width less than screen width, image color depth less than or equal to 24 bit, etc.

In addition, we define several vocabularies to represent the actions in removing an element in text/html document, and in changing the coding of a web page. For MIME type t, suppose the set of transcoding operations for type t is $\mathbf{O_t}$. The union of the above sets forms the set of all statements of our RDF model:

$$S_{RDF} = \bigcup_t (C_t \cup O_t) \cup L \cup D$$

The preconditions of the transcoding rules is a subset of $\bigcup_t C_t \cup L \cup D$. The result of these transcoding rules is a sequence of operations. An example sequence is: update image size, change coding, etc. Suppose the set of possible transcoding actions for MIME type t is called $\mathbf{M_t}$. In CAPS, $\mathbf{M_t}$ is formally defined as:

$$M_t = O_t \cup C_t$$

We call $\mathbf{M_t}$ the transcoding module for MIME type t. The result of transcoding rules for web content with type t could be represented a subset of $\mathbf{M_t}$. Finally, the set of all possible transcoding rules for MIME type t could be defined as:

$$R_t = P(C_t \cup D \cup L) \times P(M_t)$$

where P( ) is the power set function. Thus, the input of the transcoding rules is a set of statements about web content ($\mathbf{C_t}$), device CC/PP configurations (**D**), and constraints (**L**). The output of the transcoding rules then is a member of P($\mathbf{M_t}$) that are the required actions for the transcoding.

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. The following example rule is to reduce the width of a JPG image file to fit the screen width of the mobile device:

Accept(DV, "image/jpeg") ^ Format(I, "image/jpeg") ^ ScreenWidth(DV,DW) ^ ImageWidth(I, IW) ^ LessThan(DW, IW) → ScaleImageByWidth(I, DW)

In the above RDF rule, DV, I, DW, IW are variables for device, image, device width, and image width. Accept(DV, "image/jpeg") is a statement of the RDF model to express that the device could render the multimedia type "image/jpeg". The other predicates before the "→" symbol could be easily interpreted similarly. So the predicates before the "→" symbol are to check the conditions about web content (Format and ImageWidth), device CC/PP configurations(Accept), and constraints(LessThan). The predicate after the "→"symbol indicate the action to be performed: ScaleImageByWidth. The above inference rule could be expressed by the following Jena rule:

```
[ 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 above rule is named ScaleImageByWidth. In Jena rules, names prefixed with '?' are variables. The namespaces "system", "ccpp", and "Content" point to 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 width of the image as width of the device, and set height of the image proportionally with respect to the adjustment ratio of the width.

6. Transcoder of CAPS

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 actions and parameters produced by Jena, it performs detailed 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.

According to file type of the web content, the Transcoder dispatches them to the corresponding adaptation component. To perform the required content transcoding operations, it will query the inferred RDF model through Jena's 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#
```

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. Implementation of CAPS

We implement the content adaptation proxy server in the Fedora Linux 13 operating system by the Java Language J2SDK 1.6.0. We use the package org.w3c.dom for handling XHTML DOM trees of the web pages, and use the java.awt.image.BufferedImage package for handling the JPEG, PNG, and GIF image files.

7.1 Implementation details

The Semantics Extractor is implemented by a Java interface, a factory for semantic extraction, and one class for each supported file type. The obtained semantic attributes for the implemented MIME types are listed in Table 1.

MIME type	Semantic attributes extracted
image/jpeg	Image Height Image Width Image Color Depth Image Format
image/png	
image/gif	
text/html	Encoding of the web page Number Of Words In line Document Type

Table 1. The extracted Web content attributes in CAPS

The transcoding rules for images are listed in Table 2, and some of the transcoding rules for HTML pages are listed in Table 3.

The transcoding modules are responsible for receiving the transcoding parameters and performing the actual content adaptation. In CAPS, we have implemented modules for HTML, JPEG, PNG, and GIF files. The JPEG, PNG, and GIF image files are handled by the java.awt.image.BufferedImage package, while the HTML files are handled by the org.w3c.dom package. The web content extraction and parsing component obtains the requested content from the internet, and use JTidy to reformat the web page into the XHTML format and then build the DOM tree for the page. Most of the transcoding modules are implemented by built-in Java classes. The only module that we do use non-built-in Java classes are for the transcoding of images to ASCII files, which is completed through open source tool Asciizer[2].

7.2 System test of CAPS

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 4.

[2]Asciizer, http://asciizer.sourceforge.net

Transcoding Rule	Comment
[ScalingImageByWidth: (system:Content content:Width ?image_width), (system:HardwarePlatform ccpp:DisplayWidth ?display_width), lessThan(?display_width, ?image_width) -> (system:Content content:ScaleImageByWidth ?display_width), (system:ScaleImageByWidth system:TranscodeType "text/jpeg")]	Adjust image width to fit in screen size
[ExtractColor: (system:Content content:Width ?image_color_depth), (system:HardwarePlatform ccpp:ScreenColorDepth ?device_color_depth), lessThan(?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.
[JpegToHTML: (system:Content content:Type "image/jpeg"), (system:SoftwarePlatform ccpp:CcppAccept ?Bag), noValue(?Bag ?li "image/jpeg"), (?Bag ?li "text/html") -> (system:Content system:TransformTo "text/html"), (system:TransformTo system:TranscodeType "text/jpeg")]	Transform JPEG to HTML.

Table 2. Transcoding rules for image files

Figure 4 shows the upper part of the tested web page (http://www.amazon.com) in a Microsoft IE 8 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.

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.
[TransformToPlainText: *(system:SoftwarePlatform ccpp:CcppAccept ?Bag),* *noValue(?Bag ?li "text/html"), (?Bag ?li "text/plain")* *-> (system:Content system:TransformTo "text/plain"),* *(system:TransformTo system:TranscodeType "text/html")]*	Transform HTML to plain text.

Table 3. Transcoding rules for HTML files

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

Table 4. Specifications and restrictions of tested mobile devices

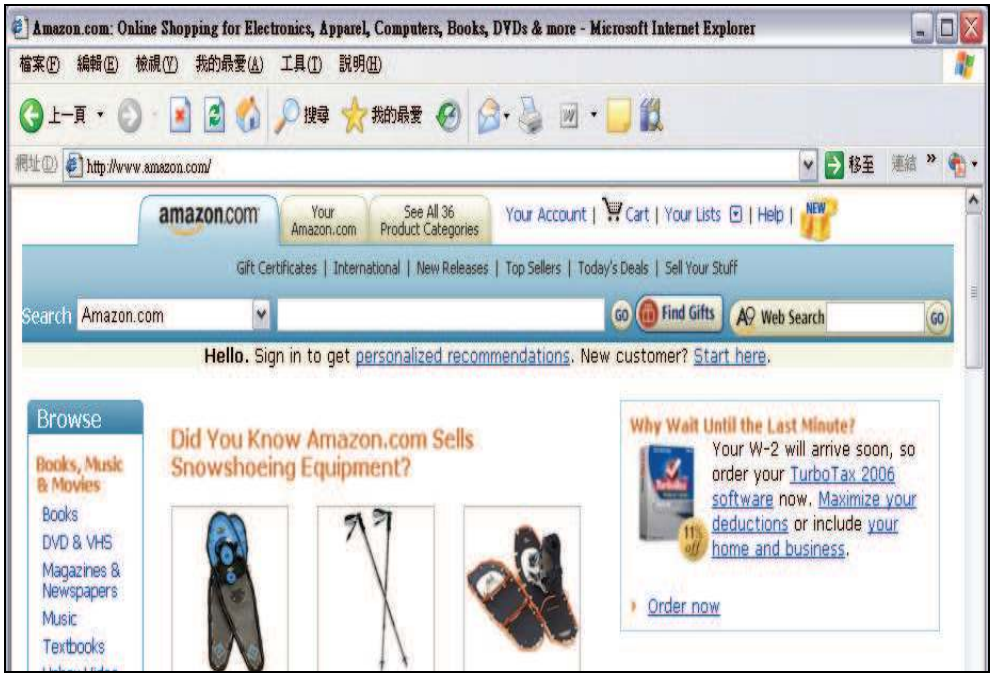


Fig. 4. Test web page in a desktop computer

Fig. 5. Results of the test page in HP iPAQ hx2400

Fig. 6. Results of the test page in Symbian Series80

Fig. 7. Results of the test page in Panasonic EB-X700

7.3 Comparison of CAPS with related work

In Table 5, we compare CAPS with related works mentioned in Section 2.1 in the following aspects: purpose, implementation levels, server deployment, dynamic adaptor loading, and transcoding parameter selection method. Some results are from the comparison about proxy-based web content adaptation system in Endler et al. (2005).

In the aspect of purpose, systems focused on Quality-of-Service [QoS] would emphasize on the reduction of transmission time and on improving users' browsing experiences. Systems focused on multimedia would emphasize on the handling of multimedia files, like summary of sound and images, and reduction of sampling frequencies. Systems with general purpose would have a flexible framework and emphasize on the dynamic deployment capability.

To handle the fast growing number of accepted web content types in mobile devices, web content adaptation systems normally would be equipped with the dynamic deployment of

transcoding modules to improve the scalability of the system. Only MARCH and CAPS provide dynamic adaptor loading of transcoding modules. In MARCH, the transcoding server was dynamically determined by the dynamic transcoding path, where each node represents a transcoding server. After traversing all nodes in the path through message passing, in which web content is encapsulated in HTTP/1.1, the web content transcoding is also finished. In CAPS, the dynamic adapter loading is implemented through Java's dynamic binding and Run-Time Type Identification [RTTI]. The transcoding modules are decided on-the-fly and thus require less loading than MARCH.

Systems	MARCH (Ardon et al.)	Lum and Lau (2002)	MoCA (Sacramento et al., 2004)	Nagao et al. (2001)	CAPS (this study)
Purpose	universal	QoS for multimedia	universal, caching	multimedia	universal
Implementation level	application	-	middleware	application	application
Server deployment	server-driven, proxy-based	-	server-driven	proxy-based	proxy-based
Dynamic adapter loading	yes	no	yes	no	yes
Transcoding parameter selection method	rules	heuristics	rules	heuristics	rules

Table 5. Comparison of CAPS with related work

In parameter selection, there are two approaches: heuristics and rules. The benefits of heuristics-controlled parameter selection are easy to be implemented in programs and faster response time. It drawbacks include less number of handled cases, and less flexibility. Once logic used in parameter selection is changed, the programs must be re-compiled. On the other hand, use of rules in parameter selection provides better flexibility. Additionally, with the capability of reasoning, more transcoding strategies could be obtained than heuristics. However, the drawback of rules-controlled parameter selection is the difficulty to write correct and proper rules.

CAPS uses light-weight transcoding components through flexible API. It could be easily deployed as distributed processing by RMI or Web Services. Therefore, the costs for deployment, allocation, and scalability could be greatly reduced.

8. Conclusion

We designed and implemented a flexible and robust framewrok, called CAPS, 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.

In CAPS, 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/Mobile (Schubert & Berjon, 2008) and XHTML for Mobile (McCarron et al., 2010), into this semantics-based content adaptation framework. By supporting these dynamic web pages, the increased user interactivity would accelerate user acceptance of pervasive computing.

9. Acknowledgment

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

10. References

Ardon S.; Gunningberg P.; Landfeldt B.; Ismailov Y.; Portmann M. & Seneviratne A. (2003). MARCH: a distributed content adaptation architecture. *International Journal of Communication Systems*, Vol. 16, No. 1, pp. 97-115, ISSN 1074-5351

Bickmore T. & Girgensohn A. (1999). Web page filtering and re-authoring for mobile users. *The Computer Journal*, Vol. 42, No. 6, pp. 534-546, ISSN 1460-2067

Butler M. (2001). Current Technologies for Device Independence, *HP Laboratories Technical Report*, No. 83

Butler M. (2002). DELI: A Delivery context library for CC/PP and UAProf, *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, *Proceeding Of the 2001 SIGCHI Conference on Human Factors in Computing Systems*, pp. 213-220, ISBN 1-58113-327-8

Carroll J.; Dickinson I.; Dollin C.; Reynolds D.; Seaborne A. & Wilkinson K. (2004). Jena: implementing the semantic web recommendations, *Proceedings of the 13th World Wide Web Conference*, pp. 74-83, ISBN 1-58113-844-X

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

Endler M.; Rubinsztejn H.; Rocha R. C. A.; Sacramento V. (2005). Proxy-based Adaptation for Mobile Computing, *Technical Report*, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Available from http://ftp.inf.puc-rio.br/pub/docs/techreports/05_24_endler.pdf

Gimson R.; Finkelstein S. R.; Maes S. & Suryanarayana L. (2003). Device Independence Principles, W3C Working Group Note, Available from http://www.w3.org/TR/di-princ/

Glover T. & Davies J. (2005). Integrating device independence and user profiles on the web. *BT Technology Journal*, Vol. 23, No.3, pp. 239-248, ISSN 1358-3948

Hayes P. & McBride B. (2004). RDF Semantics, W3C Recommendation, Available from http://www.w3.org/TR/rdf-mt/

Hori M.; Kondoh G.; Ono K.; Hirose S. & Singhal S. (2000). Annotation-based web content transcoding, *Computer Networks*, Vol. 33, No. 1-6, pp. 197-211, ISSN 1389-1286

Hsiao J. L.; Hung H. P. & Chen M. S. (2008). Versatile transcoding proxy for internet content adaptation. *IEEE Transactions on Multimedia,* Vol. 10, No. 4, pp. 646 - 658, ISSN 1520-9210

Hsu I. C.; Chi L. P. & Bor S. S. (2009). A platform for transcoding heterogeneous markup documents using ontology-based metadata. *Journal of Network and Computer Applications,* Vol. 32, No. 3, pp. 616-629, ISSN 1084-8045

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, *IEEE Transactions on Mobile Computing*, Vol. 5, No. 12, pp. -1662, ISSN 1536-1233

Huang A. W. & Sundaresan N. (2000). A semantic transcoding system to adapt web services for users with disabilities, *Proceeding of the 4th international ACM conference on Assistive technologies*, pp. 156-163, ISBN 1-58113-313-8

Hwang Y.; Kim J. & Seo E. (2003) Structure-aware web transcoding for mobile devices. *IEEE Internet Computing*, Vol. 7, No. 5, pp. 14-21, ISSN 1089-7801

Klyne G.; Reynolds F.; Woodrow C.; Ohto H.; Hjelm J.; Butler M. & Tran L. (2004). Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0, W3C, Available from http://www.w3.org/TR/CCPP-struct-vocab

Le Hégaret P.; Whitmer R. & Wood L. (2009). Document Object Model (DOM), W3C, Available from http://www.w3.org/DOM/

Lum W. Y. & Lau F. C. M. (2002). A context-aware decision engine for content adaptation. *IEEE Pervasive Computing*, Vol. 1, No. 3, pp. 41-49, ISSN 1536-1268

Manola F. & Miller E. (2004). RDF Primer, W3C, Available from http://www.w3.org/TR/rdf-primer/

McCarron S.; Ishikawa M.; Baker M.; Matsui S.; Stark P.; Wugofski T. & Yamakami T. (2010). XHTML™ Basic 1.1 - Second Edition, W3C Recommendation, Avail able from http://www.w3.org/TR/2010/REC-xhtml-basic-20101123/

Nagao K.; Shirai Y. & Squire K. (2001). Semantic annotation and transcoding: making web content more accessible, *IEEE MultiMedia*, Vol. 8, No. 2, pp. 69-81, ISSN 1070-986X

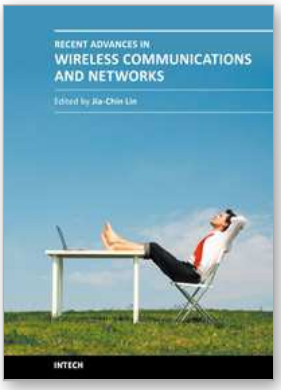
Nimmagadda Y.; Kumar K. & Lu Y. H. (2010). Adaptation of multimedia presentations for different display sizes in the presence of preferences and temporal constraints, *IEEE Transactions on Multimedia,* Vol. 12, No. 7, pp. 650 - 664, ISSN 1520-9210

Ohto H. & Hjelm J. (1999) CC/PP exchange protocol based on HTTP Extension Framework, W3C, Available from http://www.w3.org/TR/NOTE-CCPPexchange

Schubert S. & Berjon R. (2008). CSS Mobile Profile 2.0, W3C Candidate Recommendation, Available from http://www.w3.org/TR/2008/CR-css-mobile-20081210/

Wireless Application Protocol Forum, Ltd. (2001). Wireless Profiled HTTP, Available from http://read.pudn.com/downloads19/doc/comm/67224/WAP-229-HTTP-20010329-a.pdf

Wireless Application Protocol Forum, Ltd. (2001). WAG UAProf, Available from http://read.pudn.com/downloads19/doc/comm/67224/WAP-248-UAProf-20011020-a.pdf



Recent Advances in Wireless Communications and Networks

Edited by Prof. Jia-Chin Lin

ISBN 978-953-307-274-6
Hard cover, 454 pages
Publisher InTech
Published online 23, August, 2011
Published in print edition August, 2011

This book focuses on the current hottest issues from the lowest layers to the upper layers of wireless communication networks and provides â€œreal-timeâ€ research progress on these issues. The authors have made every effort to systematically organize the information on these topics to make it easily accessible to readers of any level. This book also maintains the balance between current research results and their theoretical support. In this book, a variety of novel techniques in wireless communications and networks are investigated. The authors attempt to present these topics in detail. Insightful and reader-friendly descriptions are presented to nourish readers of any level, from practicing and knowledgeable communication engineers to beginning or professional researchers. All interested readers can easily find noteworthy materials in much greater detail than in previous publications and in the references cited in these chapters.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Chichang Jou (2011). A Semantics-Based Mobile Web Content Transcoding Framework, Recent Advances in Wireless Communications and Networks, Prof. Jia-Chin Lin (Ed.), ISBN: 978-953-307-274-6, InTech, Available from: http://www.intechopen.com/books/recent-advances-in-wireless-communications-and-networks/a-semantics-based-mobile-web-content-transcoding-framework

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821