An Adaptation System based on Personalized Web Content Items for Mobile Devices

Su-Do Kim[1] and Man-Gon Park[2]
[1] Dept. of Information Systems,
Pukyong National University, Busan, Rep. of Korea
[e-mail: kim-sudo@hanmail.net]
[2] Division of Electronic, Computer and Telecommunication Engineering,
Pukyong National University,
Busan, South Korea
[e-mail: mpark@pknu.ac.kr]
*Corresponding author: Man-Gon Park

Received September21, 2009; revised November 30, 2009; accepted; December 7, 2009
published December 30, 2009

Abstract

Users want to browse and search various web contents with mobile devices which can be used anywhere and anytime without limitations, in the same manner as desktop. But mobile devices have limited resources compared to desktop in terms of computing performance, network bandwidth, screen size for full browsing, and etc, so there are many difficulties in providing support for mobile devices to fully use desktop-based web contents. Recently, mobile network bandwidth has been greatly improved, however, since mobile devices cannot provide the same environment as desktop, users still feel inconvenienced. To provide web contents optimized for each user device, there have been studies about analyzing code to extract blocks for adaptation to a mobile environment. But since web contents are divided into several items such as menu, login, news, shopping, etc, if the block dividing basis is limited only to code or segment size, it will be difficult for users to recognize and find the items they need. Also it is necessary to resolve interface issues, which are the biggest inconvenience for users browsing in a mobile environment. In this paper, we suggest a personalized adaptation system that extracts item blocks from desktop-based web contents based on user interests, layers them, and adapts them for users so they can see preferred contents first.

Keywords*:* Adapting system, item block, segment algorithm, mobile web, personalized web contents

DOI: 10.3837/tiis.2009.06.004

1. Introduction

Today, the internet is a basic communication media for acquiring information. Users want to access various web contents with their mobile devices in the same manner as desktops. Accessing the World Wide Web with mobile devices and using its contents is called Mobile Browsing. Many experts expect that the future of web service will be based on mobile devices [1]. Most of the current major internet services that are available for mobile devices are carrier-made and provided web contents, and are suited for mobile devices only. Due to the differences of resources between desktop and mobile devices such as hardware, software, and network bandwidth, it is hard to provide existing desktop-based contents for mobile devices. Recently, the performance and network bandwidth of mobile devices has been greatly enhanced, thus making it possible to provide the same desktop-based contents to mobile devices. It is called Full Browsing Service.

However, web contents are still optimized to be served from desktop, so when browsing current desktop-based contents with mobile devices, users feel inconvenienced because of the download time, performance, and scrolling needed to display a whole page. For example, if you open a web page optimized for a 17 inch display with a 15 inch display, you will feel the same inconvenience. Nowadays, there are many devices with a 3 inch screen size and 800x480 resolutions to provide a similar browsing environment to desktop. But at the same time, the browsing environment of desktop is also enhanced, and most of the recent contents are optimized for 1024×768 resolution, meaning that it will be hard for mobile devices to use the same desktop-based contents unless it provides exactly the same browsing environment with desktop [2]. The contents adaptation system is a transcoding method that adapts web contents to ensure they are optimized for various users' mobile devices, enabling users to use web contents that are optimized.

There have been many studies on adaption systems, and most use heuristic methods that analyze code, find certain patterns, segment them into blocks, and divide the blocks into sub pages optimized for mobile devices. But if the contents dividing basis is limited to code or size only, it will cause many problems. The web contents consist of many detailed items such as menu, login, search, news, shopping, etc. Among these items, users will find items which they are interested in and click hyper links of items. If an item gets divided into several blocks, the user will find it difficult to recognize the whole content and will need to navigate several blocks to fully understand a single content. Especially in case of complex contents with many items such as portals, the basis of block dividing will be a critical issue. Also, the most inconvenient feature of mobile browsing is the interface [2][3]. Mobile devices provide a limited screen size compared to desktop, and the number of queries displayed in one screen is limited also. Users require the proper interface to find their frequently used queries. Personalized service, which is mostly used in desktop, searches and suggests user favored items, and it is a very useful service assisting user searches and lessening interface inconvenience[1][4][5].

This paper will suggest a personalized web adaptation system, which will extract blocks based on items through contents analyzing of web contents, layer them, use device and preference information to create personalized web contents, reorganize them by assigning priorities to items based on user preference information, and adapt the styles for the user's device.

The rest of the paper is organized as follows: Section 2 summarizes the related research work. In Section 3, we present the architecture of our personalized web adaptation system. Section 4 presents our experimental design and result analysis. Section 5 summarizes the themes and identifies some future research directions.

2. Related Work

2.1 Mobile Browsing

Recently, mobile devices have been considered as portable computers rather than just mobile phones. Displaying desktop-based web pages with mobile devices is called mobile browsing. There are several methods to display web pages in mobile browsing, which are, displaying it as it is without any reducing or magnifying screen, using a mini map of a whole screen, using a thumbnail which shows a minimized image of a whole screen and magnifies sections when it is selected, and optimizing pages to the width of the screen and aligning components alongside the height. Recent mobile devices can be classified into several types by their purposes; service, resolution, performance, or others. Mobile devices were designed for voice communication initially, but now they have been improved, for example smart phones, which are called "Mobile PCs". This is because users want to access web contents anywhere and anytime with their mobile devices.

There is currently no method to show web contents in exactly the same manner as desktop for mobile devices. There have been many studies on adaptation systems aimed at creating optimized web contents for mobile devices. Among several methods, the height-aligning method is considered optimized for mobile devices, because of its readability, and it is widely used in Europe [2][4].

But in case that adaptation is based only on the width of the screen, users should use scrolling or navigate among the pages, and it may divide the same contents into several sub-contents that are hardly recognizable to users. Unlike desktop, with mobile devices which have an inconvenient interface, these kinds of problems can increase the inconvenience to the user. Providing personalized web contents based on user preference can show users favored items first in order to lessen the inconvenience of searching [3][5].

2.2 Personalized Service

Mobile devices usage is based on users' personal characteristics as well as their cultural backgrounds. Mobile browsing should be able to provide optimized web contents for limited mobile environments and the different situations of each mobile user [3][5]. To enable mobile devices to display desktop-based web contents, the service provider should know all of the characteristics of different mobile devices. CC/PP (Composite Capability/Preference Profiles) is an RDF format file which describes the mobile device information. It is a W3C standard to provide information such as the properties of hardware, software, network, and application, so servers or contents providers can satisfy user requirements [6][7][8].

User preference profiles can be a means to provide personalized service for users who use different devices in terms of providing personalized and individualized user experiences. Anderson and others mention the issues below as reasons for providing personalized web contents for mobile device users [1][5]:

- To make it easier to find a frequently visited destination,
- To emphasize interesting contents for users, and

- To eliminate non-interesting contents and structure.

The user preference profile becomes the key of a suggestion system for personalized web contents that collects and saves user feedback about an item from a given domain, and decides which item should be suggested.

2.3 Block Segment Algorithm

To display web contents with mobile devices having significantly limited screen size, memory, and network bandwidth, many studies have been done on the segment algorithm, which extracts blocks proper for mobile devices. There are block segment algorithms to divide web contents into blocks proper for mobile devices and deal with concerns about the various limitations of mobile devices.

One of the contents dividing algorithms is the Structure-Aware Web Transcoding Algorithm that was suggested by Y. Hwang, J. Kim, and E. Seo. It is focused on preserving the original structure of web contents for extracting blocks [8]. This algorithm finds duplicated layout patterns and transforms them into sub-pages so it can preserve a basic structure such as a table. **Fig. 1** shows a screen capture after applying Structure-Aware Web Transcoding.

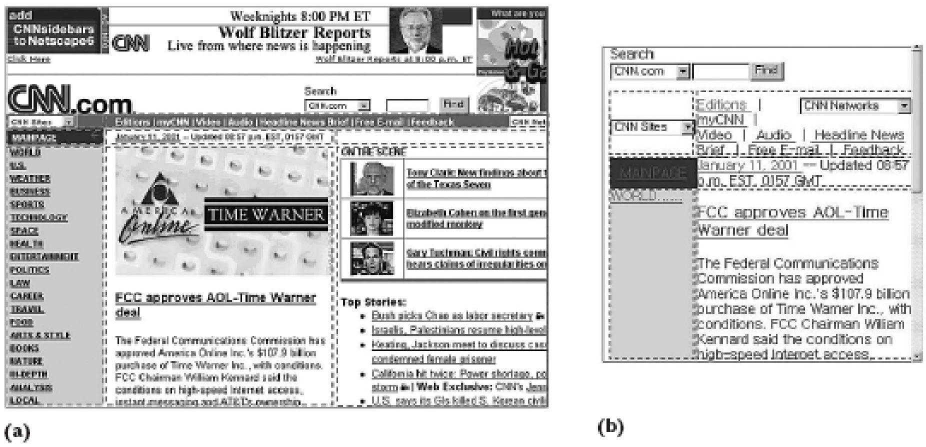

Fig. 1. (a) CNN Website (b) CNN Website after applying Transcoding

The Vision-based Page Segmentation Algorithm suggests a segment algorithm based on a visual trace system which is similar to the visual recognizing system of a human [9][10][11][12]. After transforming web pages into a DOM structure using a 3-step segmentation process of extracting visual blocks, extracting visual dividers, and constructing contents structure, it acquires the vision-based contents structure. In the block extracting process, it extracts blocks based on visual traces, assigns a DoC (the Degree of Coherence) point to each block in the form of an integer from 1 to 10, and then puts the blocks into a pool for visual separator detection. In the divider extracting process, there is an extract divider which can divide extracted blocks, apply weights, and merge blocks by the weights in order to construct the vision-based structure. **Fig. 2** shows the process of the VIPS Algorithm.

Yu Chen and others suggested an algorithm which classifies web contents blocks into five categories, re-segments blocks by size, and links each block to their sub pages in thumbnail image format [13][14]. **Table 1** shows the steps of Block Segment creation.

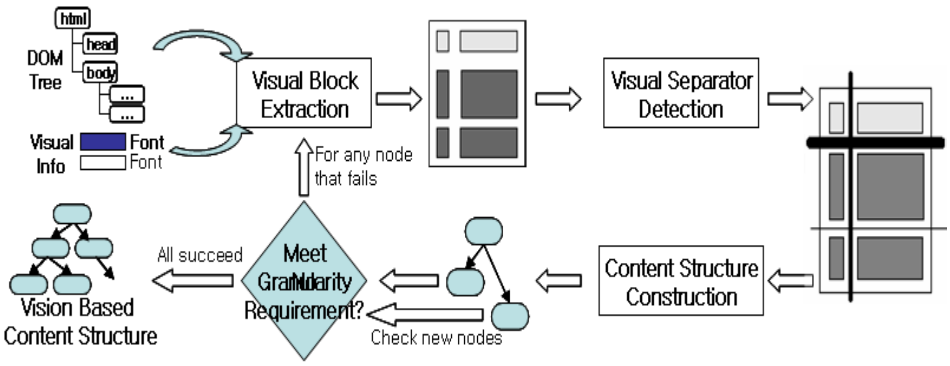

Fig. 2. Process of VIPS Algorithm

Table 1. Steps of Block Segment creation

Index	Step
1	Extracting High-Level Contents Blocks (headers, footers, left sidebar, right sidebar, body) after creating the Document Object Model (DOM) tree in the HTML source.
2	Analyzing Contents in Each High-Level Contents Block and Recognizing Explicit Separator for Block Segments
3	Detecting Implicit Separator for detailed segment of Blocks

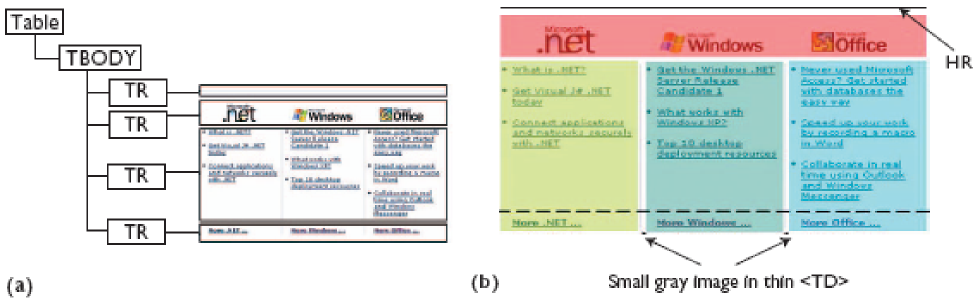

Fig. 3. Partitioning: (a) The third node in the body block, (b) Looking for explicit separators to further partition the high-level content block.

2.4 Mobile Adaptation System

Several adaptation systems have been developed and suggested. They consider users' various mobile device environments for automatically adapting desktop-based web contents to provide service. MobileGate suggested a service which converts all images into mobile-displayable images to prevent non-mobile-displayable contents. It converts a web page into an image and configures into mobile web page suitable for personal terminals using a caret unit play method [15]. Jiang He and others suggested the Xadapter adaptation system, which uses a rule-based approach for ensuring flexibility and expandability [16]. Xadapter describes an adaptation technology for content types such as text, image, streaming media, and various structures, and especially structural adaptation technology such as tables and frames. MobiDNA is focused on dynamic web contents which are changed dynamically. It is based on the adaptation algorithm of Yu Chen, and suggests a method to utilize information fragments to adapt dynamic web contents and store adapted contents into a mobile user cache that reduces mobile network bandwidth and adaptation cost [13][14][17]. Stephen J.H. Yang and others suggested a UOI-based adaptation technology, which analyzes meaningful coherence based on multimedia objects of web contents, and extracts dynamic UOI, the atomic element expression unit [18].

3. Personalized Web Adaptation System

Some adaptation technologies have been suggested to browse web contents with mobile devices having limited display, performance, and network bandwidth compared to desktop [19][20][21][22]. The purpose of these methods is to adapt desktop-based web contents for various mobile devices, so users can browse mobile friendly web contents in the same manner as desktop computers. To adapt desktop optimized web contents to mobile devices, there are many studies using a heuristic adaptation method, which finds patterns to extract blocks by analyzing code, and segments the blocks into sub-pages. But when only code or size is used as the extraction basis, some coherent contents may be divided into several sub pages, so users may have difficulties recognizing the content and will have to navigate through links. Also, since most of the pages consist of a complex item structure - which is the minimal unit of web contents - login, news, search, shopping, etc., the segmentation result may not be satisfying, and if the segment is based only on blocks, the user must navigate through links to find needed items.

In this section, we suggest a personalized web adaptation system to solve these problems. First, we extract the minimal unit of contents – items (log in, integrated search, news, shopping, etc.) from web contents, and then construct a layered system. Second, we provide personalized and individualized service optimized for individual characteristics by repositioning items by user preference, to minimize the difficulty of navigation. Third, we adapt the style of the web contents to the user's mobile device. Finally, we minimize the number of sub-pages optimized to the mobile device, to minimize the difficulties of navigating through links.

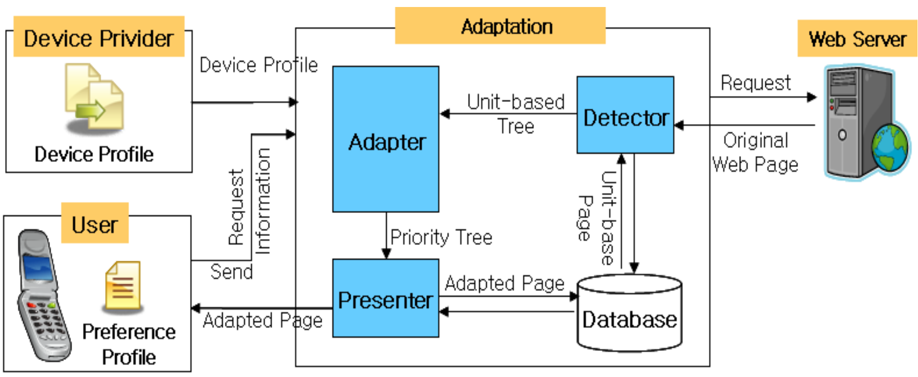

Fig. 4. Proposed Framework of Personalized Web Adaptation System

Fig. 4 shows the framework of the item-based adaptation system for personalized web contents proposed in this paper. The framework consists of three modules (Detector, Adapter, and Presenter). The system is implemented using the JAVA language.

The Detector module creates the DOM tree from the web contents-based desktop and extracts item blocks through semantic inference. The Adapter module calculates priority values of item blocks using position and interest values and reorganizes blocks by priority values. Web contents include various web languages such as JavaScript, Style Sheet etc. The Presenter module adapts the properties and styles of item blocks by the user device profile and divides them into sub pages.

3.1 Detector Module

Recent web pages have been designed to have more complex structures to display lots of items in one screen. For these web contents, the basis of recognizing a screen is the items which classify each of the contents. Especially for web contents which have many items with a complex structure such as a portal, an algorithm to extract blocks based on items is needed. The detector module uses the item-based segment algorithm proposed in [23] that extracts block and constructs a layered system by using HTML code analysis and semantic relation analogy.

Table 2. Rules to extract item blocks [23]

Index	Name	Rules
1	Invalid Rule	One of two unit blocks is an invalid block
2	Low-Text Rule	One of two unit blocks includes only the text and the text length is less than 50
3	Form Rule	Two unit blocks are included in the same form block
4	Image Rule	One of two unit blocks is a image block
5	Head Rule	Previous unit block is a head block and next unit block is not a head block
6	Same-Pattern Rule	Two unit blocks have a similar pattern structure

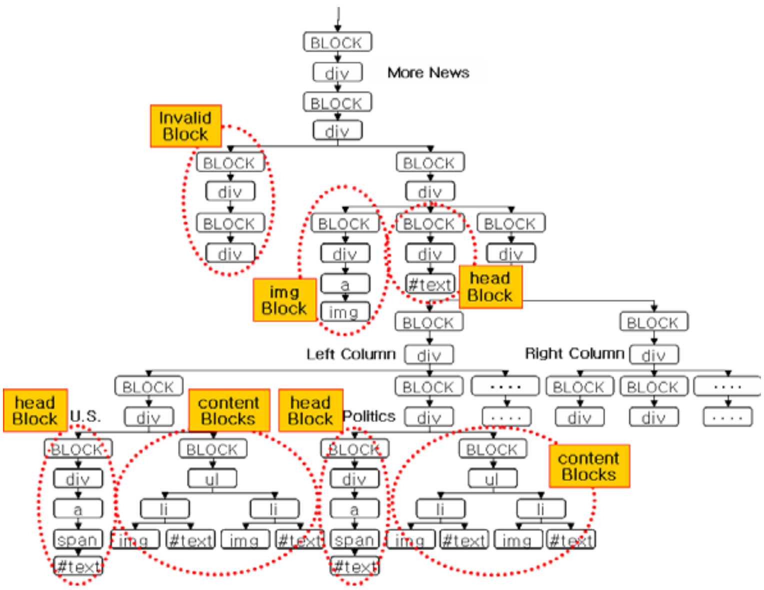

Fig. 5. Unit Blocks of “More News” in CNN Web contents [23]

The item-based segment algorithm is a three stage HTML process of parsing, extracting blocks, and extracting items. At the HTML parsing stage, the process reads the HTML source from the web server and creates the DOM tree. At the extracting block stage, it analyzes the

HTML nodes of the DOM tree and restructures the DOM tree to unit block trees. At the extracting item stage, it analyzes the semantic relations between blocks and merges them if they fall under the item rules to extract item blocks. **Fig. 6** shows a screen capture of the extracted item blocks after applying the rules.

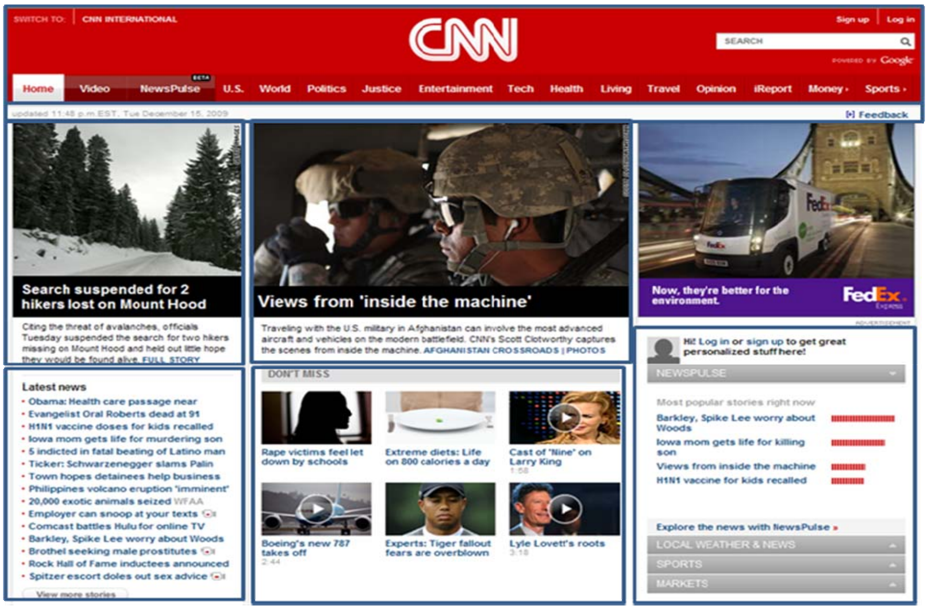

Fig. 6. Extracted Item Blocks from CNN Web contents

3.2 Adapter Module

3.2.1 User Profile

We will use the CC/PP-based user profile to provide personalized web contents adapted to the user's environment. The user device profile contains the information of the user's mobile device as the UAProf type - one of the CC/PP profiles - and the user preference profile contains the user's preference information on each item [6][7][21]. **Fig. 7** shows the mobile device sample of SU-7r10 of Nokia saved as the UAProf type.

We will use the user preference profile, which was created by item-based collaboration filtering, to ensure that frequently used (user preferred) items, are displayed first on the screen. This lessens the user's movement to find items. In this paper, we will assume that the user preference information is stored in the profile by the item-based collaboration filtering method.

Fig. 8 shows the user preference profile. ItemID is the ID of each item and is not a duplicated key. All information of ItemID is stored in the database and matched with profile properties. VisitedCount is the user access count and LastDate is the date when the user last accessed the item.

3.2.2 Priority Value

Under the block level, in order to redeploy item blocks that are extracted from the detector module, we use the position and priority values to assign the Item Priority. First, it will detect the item type of the extracted item and it will find the text or property related to the title of block and compare with the item information table stored in the database, to determine the type of item block. **Table 3** represents the steps to detect the type of item block.


```
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:prf="http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppsch
   -20021212#"
   xmlns:mms="http://www.wapforum.org/profiles/MMS/ccppschema-
   20010111#">
 - <rdf:Description rdf:ID="Profile">
  - <prf:component>
   - <rdf:Description rdf:ID="HardwarePlatform">
      <rdf:type
       rdf:resource="http://www.openmobilealliance.org/tech/profiles/UAPRO
       -20021212#HardwarePlatform" />
      <prf:BitsPerPixel>12</prf:BitsPerPixel>
      <prf:ColorCapable>Yes</prf:ColorCapable>
      <prf:CPU>ARM</prf:CPU>
      <prf:ImageCapable>Yes</prf:ImageCapable>
      <prf:Keyboard>Disambiguating</prf:Keyboard>
      <prf:Model>SU-7</prf:Model>
      <prf:NumberOfSoftKeys>0</prf:NumberOfSoftKeys>
      <prf:PixelAspectRatio>1x1</prf:PixelAspectRatio>
      <prf:PointingResolution>Character</prf:PointingResolution>
      <prf:ScreenSize>320x240</prf:ScreenSize>
      <prf:SoundOutputCapable>No</prf:SoundOutputCapable>
      <prf:TextInputCapable>No</prf:TextInputCapable>
      <prf:Vendor>Nokia</prf:Vendor>
      <prf:VoiceInputCapable>No</prf:VoiceInputCapable>
     </rdf:Description>
    </prf:component>
  - <prf:component>
   - <rdf:Description rdf:ID="SoftwarePlatform">
      <rdf:type
       rdf:resource="http://www.openmobilealliance.org/tech/profiles/UAPRO
       -20021212#SoftwarePlatform" />

       <prf:AcceptDownloadableSoftware>No</prf:AcceptDownloadableSoftware>
    - <prf:CcppAccept>
     - <rdf:Bag>
         <rdf:li>image/gif</rdf:li>
         <rdf:li>image/jpeg</rdf:li>
```


Fig. 7. Mobile Device Profile of Nokia SU-7r10 Model


```
<?xml version="1.0" ?>
- <rdf:RDF xmlns="http://www.w3.org/1999/02/22-rdf-
   syntax-ns#"
   xmlns:user="http://semi.pknu.ac.kr/user_profiles/"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
   ns#">
 - <rdf:Description rdf:about="user">
    <user:userId>ksd</user:userId>
   </rdf:Description>
 - <rdf:Description rdf:about="item#C">
    <user:itemId>C02</user:itemId>
    <user:itemVisited>2</user:itemVisited>
    <user:lastAccessed>31.03.2008</user:lastAccessed>
   </rdf:Description>
 - <rdf:Description rdf:about="item#C">
    <user:itemId>C04</user:itemId>
    <user:itemVisited>10</user:itemVisited>
    <user:lastAccessed>10.05.2008</user:lastAccessed>
   </rdf:Description>
 - <rdf:Description rdf:about="item#F">
    <user:itemId>F01</user:itemId>
    <user:itemVisited>2</user:itemVisited>
    <user:lastAccessed>10.08.2007</user:lastAccessed>
   </rdf:Description>
 - <rdf:Description rdf:about="item#M">
    <user:itemId>M02</user:itemId>
    <user:itemVisited>12</user:itemVisited>
    <user:lastAccessed>14.03.2008</user:lastAccessed>
   </rdf:Description>
 - <rdf:Description rdf:about="item#M">
    <user:itemId>M01</user:itemId>
    <user:itemVisited>15</user:itemVisited>
    <user:lastAccessed>31.04.2008</user:lastAccessed>
   </rdf:Description>
```


Fig. 8. User Preference Profile Sample

Table 3. Steps to detect an item block's type

Index	Step
1	Finding a title in an item block ▪ Text of Head tag : <H1> ~ <H6> ▪ "head" or "title" properties of CLASS or ID in tag ▪ Text of a tag, kind of TEXT tag : <strong>, <b>, <em>, etc.
2	If a title includes ItemName of the item information table stored in the database, then assign the item type of ItemName to the item block's type
3	If a title includes ItemKeyword, then assign the item type of ItemKeyword to the item block's type
4	If the content of the item block includes ItemKeyword, then assign the item type of ItemKeyword to the item block's type
5	If there is no match in the above steps, then assign the default type to the item block's type

Web developers put items which they want to show or are most frequently accessed by users at the top left corner so they can be easily found first. So an item at the top can be considered more important than one at the bottom [1]. Therefore, Position Value assigns priority in ascending order from the first-to-last item. **Fig. 9** shows the position value of the items.

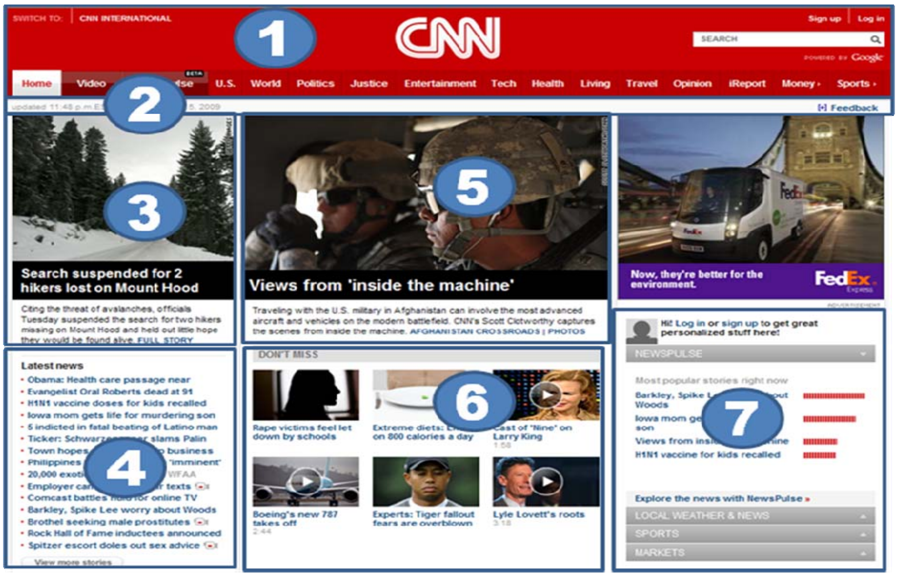

Fig. 9. Item Position

Most people tend to prefer an item that is similar to one they have previously favored and dislike an item that is similar to one they have previously disliked. We read the access count and last accessed date of an item that matches an ItemID of an item block from the user preference profile in order to calculate the interest value. It is set to 0 if there is no matched information. The item's interest value is determined by dividing the access count by the time that has passed since the last access, as shown below.

$$[\text{Interest Value}] = [\text{VisitedCount}] / [\text{days since LastDate}] \quad (1)$$

The priority of each item in the web contents can be determined by the position and interest values in the web page.

$$\text{Priority} = \text{Position Value} + (-\ \text{Interest Value}) \quad (2)$$

Position Value is assigned in ascending order based on position. But Interest Value is assigned in descending order based on preference. Then the priority value is found by adding the position value to the negative interest value. The item block that has the smallest priority value will be shown at the top of the page. To reorganize the extracted item blocks by item priority, we use a search block-based tree structure for the item block and put it in the item list. We use insert sorting to sort the items in ascending order of priority and re-create the tree by the order of item blocks inserted into the item list.

3.3 Presenter Module

To decorate items of web contents, the style sheets (CSS) language is used as well as HTML tags. But when desktop-based style sheets are applied, using the same sheet for a mobile device will generate problems. We will explain the presenter module that processes the style of the item block - the result of the adapter module - and sub-pages.

The presenter module settles, among the style properties, those that are affected by device environments, such as the position that determines the position of an item, and the width that determines the width of an item. We leave the other style properties as they are. Also, since the performance of the mobile device is significantly limited compared to desktop, if the web pages are transmitted as they are, the delay may be excessive or the page may not be displayed in the worst case, due to the size problem. However, if it is divided into too many sub-pages the user may have difficulties navigating through each page, so we calculate the size of each item block and minimize the number of sub-pages.

3.3.1 Item Style Adaptation

Most desktop-based web contents use the style sheet language as well as HTML tags to design the shape of items. But these style sheets are optimized to desktop, so in many cases they may not be displayed properly in mobile devices. Therefore, we analyze and adapt the properties of HTML tags and style sheets for the mobile environment. Some styles need to be removed and others need to be adapted.

Some web developers create a separate style sheet file and link to it. For example, among the children of the <HEAD> node, they may link to a style sheet file at <LINK> tag, or use <STYLE>@import and </STYLE> to import a style sheet. The presenter module traces and reads all related style sheets, inserts contents of the style sheets into the file, and removes the @import property of the <LINK> or <STYLE> tags to adapt the style sheet.

Table 4. Style Properties to Remove

Property	Function
Position	Assign a position of tag in screen
Margin	Assign a space of tag margin-top, margin-bottom, margin-left, margin-right
Padding	Assign a gap of tag and text padding-top, padding-bottom, padding-left, padding-right
Spacing	Gap letter-spacing
Top, Bottom, Left, Right	Assign a position of tag

Table 5. Style properties to adapt

Image	1. When width of image is larger than width of mobile screen • Change image width to mobile width • Image height = [image height] * ([image width]/[mobile width]) 2. When width of image is smaller than mobile width • Assign original image width
Width	Find width properties of style sheet or width properties of tag. If they are larger than the width of the mobile screen, then they are adapted to the width of the mobile screen

3.3.2 Page Segmentation

If the file size is excessive it also reduces the speed. Page segmentation based on device environment is needed. But excessive segmentation into sub-pages will result in excessive navigation. Many developers think that web pages should be as light as possible (each page should contain as little information as possible) but in fact the page should contain as much information as possible. For example, in a scroll-selecting device, scrolling through a nine item list is not much harder than scrolling through a 4-5 item list [1].

Each item consists of multimedia items as well as texts. The size of texts can be calculated by using the getBytes() method, and for multimedia files, we can measure the file size in order to calculate the size of the item block. If the file size is larger than the criteria, then it is divided into sub-pages.

4. Experimental Classification Results and Analysis

4.1 Simulation

We applied and tested the item-based adaptation method on the index page of the CNN website chosen from among many desktop-based web sites. The proxy for the personalized web adaptation test had 3.2GHz CPU, 1G RAM. The user device profile for the test was the Nokia SU-7r10 mobile device with 320×240 screen and the Samsung SPH-S1300 with 176×220 screen. Also, we used two samples of preference profiles that show user interests. To derive different adapted results, Preference sample 1 was higher on the Blog and Title items of Interest Values, and Preference sample 2 was higher on the Menu and Navigation items of Interest Values.

Fig. 10 shows the screen shown in the desktop and full browsing environment of the LG-LH2300 mobile device with 3 inch (7.62 cm) screen and 800x480 resolutions. LG-LH2300 is one of the mobile devices developed to provide a similar browsing environment with desktop for Full Browsing Service. Some users can be hard to read, because the 3 inch screen size represents the 800x480 resolution. In addition, the mobile device provides the zoom function.

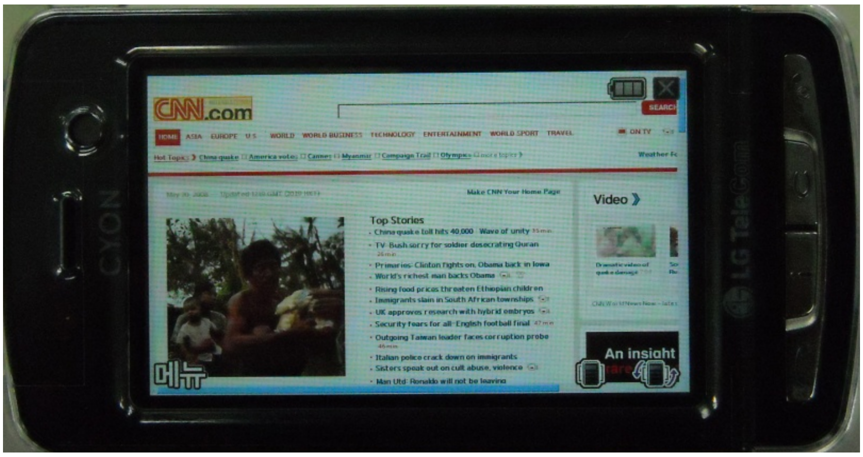

Fig. 10. Full browsing of original CNN web contents in LG-LH2300

Fig. 11 and Fig. 12 shows screen captures of the adapted web contents from each mobile device. For the simulated mobile device, we used the LG-LH2300 instead of the Nokia SU-7r100.

The web content is adapted to the same mobile device, but it shows a different item order of web contents by the preference information of the user.

(a) Preference 1 applied (b) Preference 2 applied

Fig. 11. CNN screen capture after applying the SU-7r100 device profile of Nokia

(a) Preference 1 applied (b) Preference 2 applied

Fig. 12. CNN screen capture after applying the SPH-S1300 device profile of Samsung

4.2 Performance Analysis

Table 6 is a list of web sites that were chosen from various categories, and the index page of these sites was used. This is because the index page is constructed with complex items to show the overall contents of the site. The results for 20 web sites were that the correct detection rate was 84.55% and the error rate was 15.45%. Among the error cases, when several different items were combined into one item the average was 0.6, and when one item was divided into several blocks the average was 3.55. There were more errors when dividing one item into several than when combining different items to one. If the item has a complex structure or irregular pattern, it creates errors when extracting blocks.

Table 6. List of tested web pages

Index	Name	URL	Category
1	Daum	http://www.daum.net/index.html	Portal
2	Yahoo	http://kr.yahoo.com/index.html	
3	Naver	http://www.naver.com/index.html	
4	Korea	http://www.korea.com/index.html	
5	Google	http://www.google.com/index.html	
6	Nate	http://www.nate.com/index.html	
7	Cyworld	http://www.cyworld.com/main2/index.asp	
8	CNN	http://edition.cnn.com/index.html	News
9	Chosun	http://www.chosun.com/index.html	
10	EtNews	http://www.etnews.co.kr/index.html	
11	Times	http://www.times.com/index.html#	
12	LATimes	http://www.latimes.com	
13	BBC	http://www.bbc.co.uk	Broadcast
14	KBS	http://www.kbs.co.kr/index.html	
15	MBC	http://www.mbc.co.kr/index.html	
16	Interpark	http://www.interpark.co.kr/malls/index.html	Shopping
17	Amazon	http://www.amazon.com/gp/homepage.html	
18	ebay	http://www.ebay.com	
19	JavaSun	http://java.sun.com/index.jsp	Company
20	IBM	http://www.ibm.com/us	

On average, the process time of the detector module is 0.291s,, that of the adapter module is 0.306s, and that of the presenter module is 0.962s. The total process time of all modules to adapt is 1.56 seconds. In case of a large style sheet, the presenter module takes more process time than the other modules. The file size for the adapted web contents is reduced by 13% on average compared to the original. This is because the screen size-related properties are removed from the style sheet. Nowadays, instead of using one method, a combined Client-side, Proxy-side or Server-side method for a distributed process can be used, so the downloading and processing time can differ by system.

Fig. 13 shows the number of items and adapted file size of web contents by classification. More items were detected from index pages of news classes than index pages of other sites. In general, a web page contains a number of items that can be displayed in one screen, and even if it exceeds the screen size, it is not displayed very far from the screen. This is because if the page is constructed to be larger than the screen size, the user should scroll through it to search for items. But in case of a news site, since it must show a wide variety of items in real time, it tends to use longer web pages. Also, more items were detected. At the index pages of shopping site, the products can be arranged and displayed by categories. Same kind of items can be classified and displayed in the same space, and some specific items are presented on a large space of the page for advertisement, therefore small number of items can be detected. **Fig. 14** shows the correct rate, error rate, and total processing time of modules by classification of web contents. **Fig. 15** shows the original file size and processing time of each module. The processing time tends to be dependent on the file size rather than the number of items, and among the files, it is dependent on the size of the style sheet. Style sheet adaptation takes more time.

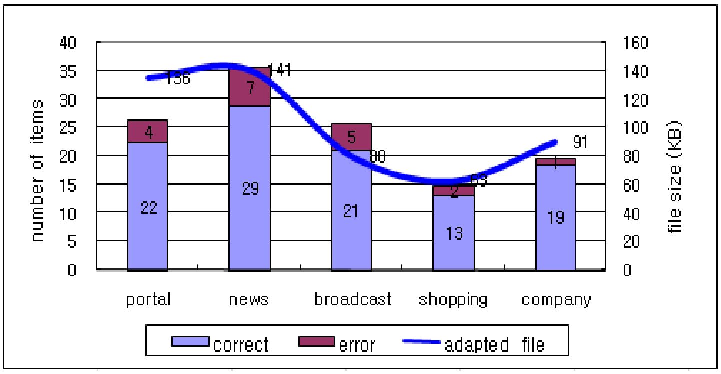

Fig. 13. Number of items and file size in categories

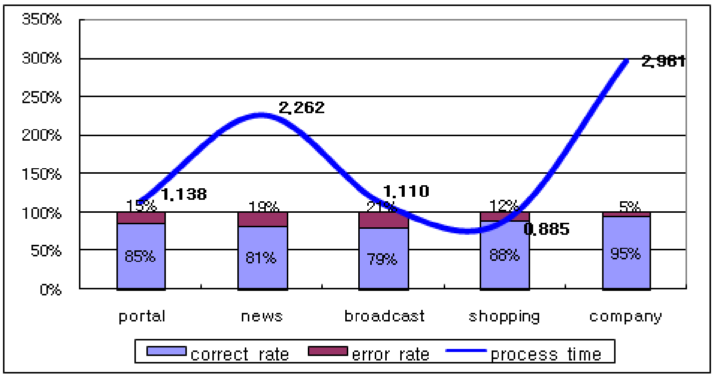

Fig. 14. Correct rate and process time in categories

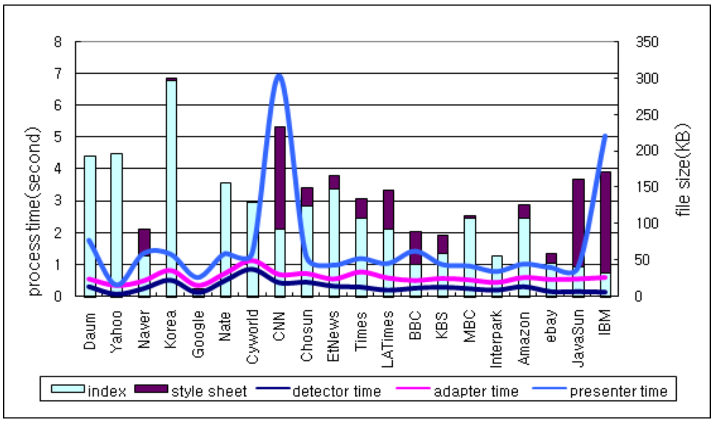

Fig. 15. File size and process time

5. Conclusions

Due to improvements of mobile devices, it has become possible for users to exchange various contents and process multimedia with them. Now desktop-based web contents can be accessed with mobile devices, so mobile browsing that can access web contents with any mobile device anywhere and anytime is possible. But the limitations of mobile devices, such as network bandwidth, device performance, screen size, and supporting software, may cause many problems when browsing desktop-optimized web contents with mobile devices. To automatically adapt desktop-based web contents to various mobile devices, many studies have been done on various adaptation methods, some of which have already been realized. But the recent web contents such as portals are getting more complicated in terms of both the structures and contents as compared with the past ones.

Most adaptation methods are based on heuristic methods to find certain patterns by code analyzing, to segment web pages into blocks and summarize and link them at the index page. If the segment basis is limited only to the size or HTML code, it becomes hard to extract complex items as one block. If an item is divided into several blocks or different items get combined into one block, it gets harder for users to recognize the item. Users recognize their favorite items by separating items when they see a page with many items. In this paper, we divided web contents by the item blocks - the unit of contents - instead of the file size, in order to segment them. We suggested a personalized adaptation method that uses the UAProf device profile information and a user preference profile containing user interest information in order to calculate the item priority and reconstruct the web tree. We adapt a desktop-based style sheet to the mobile device, and divide pages optimized for it.

For future work, we are planning to improve our system in three directions. First, we will apply a more accurate priority value that is recomputed by the theory of classification scheme. Second, we will find various factors important for providing appropriate web contents to the user, such as usability, reliability availability, serviceability etc. Third, we will study the applying method of machine learning to improve personalization.

We expect that mobile devices will be the basic device of importance for the future of the web, not just an alternative when desktop is not available. Personalized web adaptation methods will be complemented and studied further. They will become a Ubiquitous-based technology which can provide users with convenient web access using any device.

References

[1] Muriel Bowie, "Adaptation of a Webshop for Mobile Devices," *Master Thesis*, Dept. of Computer Science, Fribourg University, Oct. 2005.

[2] V. Roto, "Browsing on Mobile Phones," *Nokia Research Center*, Retrieved from http://www.research.att.com/~rjana/WF12_Paper1.pdf, Sep. 2009.

[3] V. Roto, "Web Browsing on Mobile Phones-Characteristics of User Experience," *Doctoral Dissertation*, Dept. of Computer Science and Engineering, Helsinki University, 2006.

[4] H. Lam and P. Baudisch, "Summary Thumbnails: Readable Overviews for Small Screen Web Browsers," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 681-690, 2005.

[5] J. Freire, B. Kumar, and D.F. Lieuwen, "Webviews: Accessing Personalized Web Content and Services," *Proceedings of 10th WWW Conference*, ACM Press, pp. 576-586, 2001.

[6] W3C, "Composite Capability/Preference Profile (CC/PP): Structure and Vocabularies 1.0," Retrieved from http://www.w3c.org/TR/2004/REC-CCPP-struct-vocab-20040115, Sep. 2009.

[7] The Wap Forum, “Wireless Application Group User Agent Profile Specification,” Retrieved from http://www.wapforum.org/what/technnical/SPEC-UAProf-19991110.pdf, Sep.2009.
[8] Y. Hwang, J. Kim, and E. Seo, “Structure-Aware Web Transcoding for Mobile Devices,” *IEEE Internet Computing Magazine*, vol. 7, pp. 14-21, Oct. 2003.
[9] D. Cai, S. Yu, J. Web, and W. Ma, “VIPS: a Vision-based Page Segmentation Algorithm,” *Microsoft Technical Report*, MSR-TR-2003-79, Microsoft Research, Nov. 2003.
[10] J. Kang and J. Choi, “Detecting Informative Web Page Blocks for Efficient Information Extraction Using Visual Block Segmentation,” *International Symposium on Information Technology Convergence*, pp. 306-310, 2007.
[11] P. F. Xiang et al., “Effective Page Segmentation Combining Pattern Analysis and Visual Separators for Browsing on Small Screens,” *International Conference on Web Intelligence*, pp. 831-840, 2006.
[12] P. Xiang, X. Yang, and Y. Shi, “Web page Segmentation based on Gestalt Theory,” *IEEE Conference on Multimedia and Expo*, pp. 2253-2256, July 2007.
[13] Y. Chen, W. Y. Ma, and H. J. Zhang, “Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices,” *Proceedings of 12th International Conference on World Wide Web*, May 2003.
[14] Y. Chen, X. Xie, W. Ma, and H. Zhang, “Adapting Web Pages for Small-Screen Devices,” *IEEE Internet Computing Magazine*, vol. 9, pp. 50-56, Feb. 2005.
[15] Dae Hyuck Park, Eui Sun Kang, Younghwan Lim, “Caret Unit Method from PC Web for Mobile Device,” *Journal of Korea Information Processing Society*, vol. 14, no. 3, pp. 339-346, 2007.
[16] J. He, T. Gao, I. Yen, and F. Bastani, “A Flexible Content Adaptation System Using a Rule-Based Approach”, *IEEE Trans. on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 127-140, Jan. 2007.
[17] Z. Hua, X. Xie, H. Liu, H. Lu, and W. Ma, “Design and Performance Studies of an Adaptive Scheme for Serving Dynamic Web Content in a Mobile Computing Environment,” *IEEE Trans. on Mobile Computing*, vol. 5, no. 12, pp. 1650-1662, Dec. 2006.
[18] S. J. H. Yang, J. Zhang, R. C. S. Chen, and N. W. Y. Shao, “A Unit of Information-Based Content Adaptation Method for Improving Web Content Accessibility in the Mobile Internet,” *ETRI Journal*, vol. 29, no. 6, pp. 794-807, Dec. 2007.
[19] I. Mohomed, J. Cai, S. Cbavoshi, and E. Lara, “Context-Aware Interactive Content Adaptation,” *Proceedings of 4th International Conference on Mobile Systems*, pp. 42-55, 2006.
[20] Eunshil Lee, Jinbeom Kang, Joongmin Choi, “Topic-Specific Mobile Web Contents Adaptation,” *Journal of Korea Institute of Information Scientists and Engineers*, vol. 34, no. 06, pp. 539-548, 2007.
[21] Su-Do Kim and Man-Gon Park, “A Study on the M-learning System on CC/PP for Multimedia Messaging Service Adaptation,” *Journal of Korea Multimedia Society*, vol. 11, no. 6, pp. 883-890, 2008.
[22] Won Seop Kim, Young Joon Chai, Tae Yong Kim, “A Visual Enhancement of Web Content Adaptation for Mobile Devices,” *Journal of Korea Multimedia Society*, vol. 10, no. 10, pp. 1229-1239, 2007.
[23] Su-Do Kim and Man-Gon Park, “A segment Algorithm for Extracting Item Blocks based on Mobile Devices in the Web Contents”, *Journal of Korea Multimedia Society*, vol. 12, no. 3, pp. 427-435, 2009.

Su-Do Kim received her B.S. degree in Computer Science, M.S. degree in Computer Science Education and Ph. D. in information systems from Pukyong National University, Rep. of Korea, in 1995, 2001 and 2008, respectively. She was a contract professor supported by the Nuri National Project from September 2008 to July 2009. And, she is a lecturer of the Department of Information Systems, Graduate School, Pukyong National University. She is interested in efficient presentation methods and performance analysis of mobile web contents for convergence technology for ubiquitous teaching and learning systems.

Man-Gon Park has been a head professor of the Division of Computer and Multimedia Engineering, College of Engineering, Pukyong National University, Republic of Korea since 1981. Also, he was the president and chairman of the Korea Multimedia Society (KMMS). He served as the Director General and CEO of the Colombo Plan Staff College for Technician Education (CPSC) from 2002 to 2007, which is an intergovernmental international organization of 29 member governments for Human Resources Development in Asia and the Pacific Region. He served there also as a faculty consultant seconded by the Government of the Rep. of Korea as an expert in information systems development and ICT-based TVET systems from 1997 to 2001. He has been the visiting professor at the Department of Computer Science, University of Liverpool, UK; exchange professor at the Department of Electrical and Computer Engineering, University of Kansas, USA; and visiting scholar at the School of Computers and information science, University of South Australia. He has been sent to Mongolia and People’s Rep. of China by KOICA on various projects as an information systems consultant. He has also embarked on consulting works and conducted training programs in ICT in an individual capacity for Korean groups of companies, governmental and non-governmental agencies and other institutions in Korea. His main areas of research are software reliability engineering, business process reengineering, Internet and web technology, multimedia information processing technology, system performance evaluation, and ICT-based human resources development.