Chapter IV
Web Content Adaptation for Mobile Devices

Introduction
Keeping security & quality of educational data, web page adaptation is essential to access the data on mobile devices. At present, various techniques of content adaptation, including automatic adaptation procedure is available. We discuss these techniques with various available open source content adaptation system and then generate a new technique to automatically adapt the contents of the website on the mobile devices. The previously available techniques of content adaptation are to adapt the contents for specific devices only, whereas our technique adapts the contents on any type of mobile device.

4.1 Web Pages and Mobile Devices
Use of Internet for educational purposes has grown to a great extent and handheld devices are being used to access Internet by both, the school children and their parents. They also use mobile devices for getting information related to their home assignments. This means the users can access web using a PC at home or at office and also can access the same information on their mobile phone away from their respective homes.

Although there are advancements in technical and bandwidth aspects, mobile devices have limitation due to their small screen sizes, which limit the amount of information to be displayed at a specific time. Mobile browsers display the content on mobile devices by two main transformation methods - direct migration and linear migration. In direct migration, no transformations are made to the original web page, while in linear or columnar approach, page areas are presented one after another in a single column. The presentation of information available on the website is changed to a long linear list that can easily fits within the small screen constraint of the mobile device. The major advantage [161] of this approach is that horizontal scrolling is not required.

Most of the HTML web pages are not supported by internet enabled mobile handheld devices because the web pages may not be properly and speedily displayed on
the micro browsers of the devices due to low memory capacity, small screen size, limited computing power, narrow network bandwidth, & resources etc. Web usage mining, a branch of web mining, can be helpful in summarizing the web pages for these devices. Web usage mining helps [59] in data gathering, navigation pattern discovery, pattern analysis etc and also helps in improving the readability and download speed of mobile web pages.

### 4.2 Types of mobile devices and their characteristics

The Evolution of mobile devices since 1979 has made us so much location independent that now we can work at any location without worrying about time, space and the resource. Today different types of mobile devices are available in the market. So it is important to know about the type of mobile device that can ease ones work and will be available with him 24x7. Presently available mobile devices may be divided into six major categories:

a) - Palm-sized PDAs  
b) - Handheld PCs  
c) - Tablet PCs.  
d) - Web-enabled phones  
e) - Low-end smart phones  
f) - High-end smart phones

To characterize the mobile devices, we have taken following parameters into consideration:

1. Type of screen  
2. Type of input  
3. Operating system they support  
4. Type of memory  
5. Processing power  
6. Type of Programming language or markup language they support

The table given below will demonstrate all the characteristics along with the mobile device with these characteristic:
<table>
<thead>
<tr>
<th>Type of Device</th>
<th>Type of screen</th>
<th>Type of input capabilities for user input</th>
<th>Operating system they support</th>
<th>Type of memory</th>
<th>Processing power</th>
<th>Type of programming language/markup language</th>
<th>Competent mobile device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Palm sized PDAs</td>
<td>160x160, 320x240 VGA, supporting colors</td>
<td>touch-screen capabilities for user input, stylus, QWERTY keyboard, voice input</td>
<td>Microsoft Pocket PC, Palm OS, Microsoft Windows Mobile SE/Premium</td>
<td>8-512 MB, usually 256 MB, additional MMC possible</td>
<td>33-624 MHz</td>
<td>HTML, XHTML, Java ME, proprietary formats</td>
<td>Palm Zire 72</td>
</tr>
<tr>
<td>Hand held PCs</td>
<td>480x320 VGA</td>
<td>QWERTY keyboard, touch screen</td>
<td>Palm OS, Microsoft Pocket PC, Microsoft Handheld PC</td>
<td>256 MB to 5 GB (internal flash drive, additional program memory for applications and data)</td>
<td>416 MHz and higher</td>
<td>HTML, XHTML</td>
<td>Samsung NEXiO</td>
</tr>
<tr>
<td>Tablet PCs</td>
<td>10.4&quot;-14.1&quot; TFT XGA</td>
<td>pen-based input, touch-screen, QWERTY keyboard</td>
<td>Microsoft® Windows® XP Tablet PC Edition</td>
<td>256 up to 2 GB SDRAM, 20 GB to 80 GB hard drives</td>
<td>from about 700 MHz up to 1.1 GHz</td>
<td>HTML, XHTML</td>
<td>Acer TravelMate 100</td>
</tr>
<tr>
<td>Web Enabled Phones</td>
<td>limited display, 4-12 lines of text (10-20 characters per line), monochrome or color</td>
<td>12-button keypad, voice input</td>
<td>none</td>
<td>2-128 MB (small to medium memory)</td>
<td>90-150 MIPS DSPs (low processing power)</td>
<td>not supported, eventually WML</td>
<td>Nokia 6510</td>
</tr>
<tr>
<td>Low end Smart Phones</td>
<td>Similar to Web-enabled phones, but equipped with Java ME Better processors, larger memory and storage facilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Nokia 5100 Nokia 7600</td>
</tr>
<tr>
<td>High end smart phones</td>
<td>640x200 to 320x240</td>
<td>keyboard, touch screen</td>
<td>Palm OS, Symbian OS, Microsoft Pocket PC Phone Edition, Microsoft Smartphone</td>
<td>typically 64 MB RAM and 64 MB Flash memory; up to 512 MB</td>
<td>about 400 MHz</td>
<td>WML, XHTML, HTML, Java ME</td>
<td>Nokia 6670 Nokia 7710 Sony Ericsson P800</td>
</tr>
</tbody>
</table>

Table 4.1 Comparison of various Mobile devices on the basis of different characteristics
In comparison to traditional desktop computers, mobile devices follow a different usage paradigm and possess certain limitations. They have a comparably lower computational power, smaller memory and cache as well as limited storage capabilities. Currently, many mobile devices use a secondary type of memory that can significantly increase the available memory capacity. It is inserted into the device as a slide-in card. Various other types of storage media are available, for example, Memory Stick, Secure Digital (SD), and Multimedia Memory Card (MMC) for this purpose.

Since the performance of a processor is directly related to the power it uses, mobile device does not have high speed processor to save its batteries. The cost factor is another reason for applying slow speed processors in wireless devices as fast processors are quite costly.

### 4.3 Adoption and Acceptance of Mobile Internet

The acceptance of mobile Internet and the availability of advanced mobile services differ widely across the world. In Europe, mobile services based on the Wireless Application Protocol did not evolved as expected since mobile Internet and mobile business still lack prevailing users’ acceptance. However, the failure of WAP / mobile Internet in Europe and its amazing success in Japan cannot be attributed solely to technical differences between protocols applied to these services in those countries. Mobile Internet was seen as a great possibility for the creation of value-added services in mobile networks. In 1997, Sony Ericsson, Motorola, Unwired Planet, and Nokia started an initiative for the development of an open standard for mobile services. In April 1998, the first Wireless Application Protocol (WAP) specification was released. I-mode was a success because of the knowledgeable strategy of DoCoMo. The company chooses a revenue sharing model in which 91 percent of the fee for using a mobile service goes to a content provider, and only 9 percent to DoCoMo. Although the operator could deliver services only at the speed of 9.6 Kbps, it introduced packet switching and charged the customers for the delivered contents. The most popular DoCoMo services were e-mail, ring tones, screensavers, and browsing.
WAP, often interpreted by its disappointed users as “Wait and Pay” or “Where Are the Phones”, is perceived by many analysts as a very disappointing experience [73]. Experts in mobile technologies and mobile solutions are of the opinion that the failure of WAP services should not be attributed to immature or insufficiently developed technology rather the WAP Forum ignored the opportunity to communicate the WAP concept to operators, content providers, and end-users. In spite of all, high expectations were created and propagated by the media. The vision could not be fulfilled because the operators from the WAP Forum failed to develop realistic business models that could encourage users and content providers. These expectations were based on the previous success of the Web and the observed tendencies on the market for mobile devices.

The research on mobile adoption emphasizes that the traditional Internet is the most dangerous rival of mobile browsing. Users consider the Internet and mobile Internet as substitutes. They go online using a mobile phone only if they are unable to access the same content using a PC (e.g. while traveling). Otherwise, they are not willing to pay for browsing with the help of a mobile device. They consider the traditional Internet as a more convenient and cheaper solution.

Users, who were asked to evaluate mobile solutions in various surveys, complain about their insufficient usability related to sensory and functional needs. They mention the following factors that discourage them from using mobile services or are obstacles in the use of wireless applications.

- Poor mobile content
- Inconvenient devices (difficult navigation and browsing)
- Insufficient download/upload quality and data transfer
- Poor stability, speed, coverage of the network
- Security aspects
- High grade of difficulty/complexity to utilize the services, problems with configuration and set up
- Too much advertising, and finally
- High costs.
Although the magnitude differ between mobile Internet users & non users, they all named similar reasons for the bad perception of mobile solutions. Both tech-savvy and less-experienced users were concerned about the technical complexity of offered applications and therefore demanded simpler, out of the-box operating experience. High costs are still an important obstacle in mobile browsing. Past surveys showed that subscribers with mobile accounts subsidized by their employers were more likely to use mobile Internet than the users who had to pay for mobile access individually. Further, Most users focus on purchasing basic voice services at the lowest price possible and do not even consider the use of mobile data services.

4.4 Essential functionality and Methods of Adaptation

The word “adaptation” originates from Latin word “adapter” (to fit to). It stands for a change in partial or entire characteristics of an entity under consideration and is always based on some context. In the context of device-independence, the term adaptation describes “a process of selection, generation, or modification that produces one or more perceivable units in response to a requested uniform resource identifier in a given delivery context” [167].

Researches on the adaptation of Web pages started with the emergence of the adaptive hypermedia concept in the 1990s. Adaptive hypermedia systems were able to adjust themselves to the needs of users. They possessed a model of user goals, preferences and his knowledge.

The adaptation was further enhanced by applying data about usage (e.g. usage frequency of particular sites or action sequences in browsing) and environment (software and hardware components, user’s location). Brusilovsky and Kobsa [121] distinguished three basic adaptation types namely adaptation of content, adaptation of presentation and modality (from image to text, from text to audio, etc.) and adaptation of structure.

With the emergence of heterogeneous devices, the necessity of adaptation became inevitable [73]. Its main goal was not to make Web-based systems more user-oriented, as in the adaptive hypermedia concept, but to enable the presentation of content on different devices. Adaptation can be characterized by its kind, i.e., changes that should be
performed, the subject of adaptation, and the quality of adaptation. With regard to the kind of adaptation, one should consider its effect, complexity, and universality. Considering the effect, adaptation may enhance an application by adding new parts to it. It may also reduce an application by removing its fragments or perform some transformations (e.g. replacing multimedia content with descriptions). Adaptation can be simple and atomic or complex, consisting of a set of adaptations. With regard to universality, adaptation may be application-specific and useful only for certain types of applications or generic and universally applicable (e.g. changing the resolution of images).

Bożena Jankowska [73] showed that the quality of adaptation can be determined by its dynamism, automation, visibility, and reusability. Adaptation may be static or dynamic. In static adaptation, the adapted content is defined at design time (e.g., to prepare two versions of the same picture for different bandwidths). In dynamic adaptation, the content is generated at application’s runtime. Depending on the user’s influence on the adaptation process, one can distinguish between manual, semi-automatic, and automatic adaptation. Manual adaptation means that the user can determine the kind of adaptation. In semi-automatic adaptation, the user can influence the adaptation process, for example, by specifying her/his preferences and choosing appropriate versions of content. Automatic adaptation is performed without the user’s interferences. Moreover, the adapted content can be visible globally or only by certain groups of users. Last but not the least, adaptation may be performed from scratch, based on the original version of content, or incrementally. In the later case, it is applied to the content transformed in preceding adaptation steps. Subjects of adaptation are style; layout, content, structure, navigation, application interactions, and transmission of data [168]. Style refers to the visual appearance of text (i.e. colors, fonts) and layout features of textual elements. Layout of content is the visual or temporal arrangement of particular components. Content is the information relevant to the users in the form of text or graphics. Structure describes relationships between parts of delivered content and is influenced by navigation that offers the user a possibility to move between presentation units. Application interaction is the manner in which the user transmits information to a server with the help of the characteristics of delivered content in order to influence subsequent content delivery (e.g.
form processing). Transmission of data refers to the method of transfer information (e.g., transportation of data processed on the client).

4.4.1 Adaptation of style and layout of documents

In the adaptation of style and layout, the contents of objects, that is tables, paragraphs, etc. remains the same, while the format and layout of the objects changes. A special type of style adaptation is the variation made to input/output modality in multimodal systems. In such systems [73], one can use alternative access methods and interact with an application in a variety of ways using speech, keyboard, or stylus. The presentation style depends upon the features of devices and on user preferences, for example, the users can specify, what colors and font types they prefer, or which arrangement of elements they find optimal. The type of device used also influence the presentation of content due to its limited capabilities to render colors or scarcity of available font types and sizes. Style adaptation is usually achieved with device-dependent style sheets like CSS or XSLT.

In layout adaptation, spatial and temporal layouts are considered. Spatial layout specifies the vertical and horizontal arrangement of areas in which content appears. Temporal layout [73] refers to voice or media applications and controls the presentation of elements in time. Furthermore, some parts of the content may be made invisible to the user.

4.4.2 Structure and navigation adaptation of documents

Aggregation and decomposition play a central role in the process of structure adaptation. In aggregation, content fragments from one or more sources are collected to form a single fragment or page [168]. Individual content fragments contain certain structures and the aggregation process leads to a new structure.

In decomposition, authored units are divided to create a new set of presentation units, suitable for a particular delivery context [168]. A decomposition approach has to specify how the units will be divided and referenced from each other (navigation mechanism). Pagination is the commonly known method for decomposition of content that possesses linear structure [168]. Different pagination techniques exist. Page breaks, window/orphan control, “keep together/keep with next”, sectioning, and regions are frequently applied in
The adaptation of navigation structure by modifying an existing structure or by creating a new one from the content is a difficult task. It can be performed automatically only in some situations where navigation paths may be determined by creating sitemap from all links or generating table of contents from headings [73]. Typical navigation structures are tables (lists), menus (hierarchies), and links (previous/next). A navigation structure may be generated by using existing links or by creating links from significant parts like titles, headings, etc. of the content. Classic navigation structures are tables of contents, navigation menus, and next and back links.

4.4.3 Content adaptation of documents

The contents of Web documents can be decomposed into three categories: text, images, and multimedia elements. Parts of documents can be adapted by changing the properties viz the resolution or color depth of an image of objects and data representations (file formats), by replacing some objects with other ones, or by eliminating content fragments. Text usually possesses some structure such as title, headings, sections, abstracts, meta information, etc. to describe its semantics. Text can be adapted by extracting only the most relevant fragments from it, or by splitting the information into many fragments, connected by an appropriate navigation structure.

In adaptive hypermedia systems, from which device-independent approaches benefit, adaptation of content, can be achieved with page or fragment variants, adaptive stretch text, fragment coloring and adaptive natural-language generation [121]. The page-variant approach is simple but inflexible and costly because for each adaptation, considered as necessary, a separate page has to be created. In the fragment-variant approach, separate variants are authored for each adaptive page fragment. The fragments can then be combined dynamically, forming one page. A frequently applied technique consists of insertion or exclusion of optional fragments, depending on user preferences or device features. A simple technique to implement the optional text approach is to associate a condition with each optional chunk of text that determines whether inclusion or exclusion should be applied to this fragment. According to Bożena Jankowska [73] in device-independent approaches, the inclusion/exclusion of content is often achieved with SMIL.
In the fragment-coloring method [121], certain elements of the presentation are marked out as being important or interesting to the users. They define as “stretch text” as the text that can be extended or collapsed by clicking on it with the mouse. Depending on user preferences, stretch text can be automatically extended for certain users. In adaptive, natural-language generation, alternative text descriptions are created for different users [73]. For example, text templates with slots can be filled with descriptions with different complexity grades, based on the user’s level of expertise.

4.4.4 Transfer adaptation

Despite recent improvements in this domain, mobile connections still rely on low bandwidth, high latency, error-prone, and expensive networks. Instead of sending the whole content to the device and adapting it on the client, documents may be adapted before the transfer. In case of network errors, the proxy can prevent possible data losses by caching the data for further use. Furthermore, transfer adaptation may take into consideration the maximal size of a WML deck or the allowed size of downloaded Java ME applications.

4.5 Device Independence Principles

To enhance user experience related to the content delivered to various devices, the W3C Consortium specified certain principles for device independence [167]. These principles should not be regarded as a strict set of requirements. They should rather serve as guidelines for the design of applications based on existing markup languages, during the development of adaptation tools, in the process of extending existing markup languages, or for the design of new markups [167]. According to these principles, content adaptation aspects can be viewed from three different perspectives: user’s perspective, author’s perspective, and delivery perspective.

User-related principles Generally speaking, the user would like to interact with the Web using various devices and via many access mechanisms. From the user’s perspective, two most important aspects of content adaptation are: device-independent access to the same functional presentation of content (DIP-1) and device independent Web page identifiers.
Device independent access means that the user is able to obtain the equivalent application functionality regardless of the device type. Obviously, the presentation will not be the same on every device and its quality may vary. The intended functionality of a page should be associated with only one Web page identifier, and should apply to all presentations obtained from the same Web page identifier, no matter what the access mechanism is. For example, a user who enters one URL for a weather service in a browser on his/her PDA and in a browser on his/her WAP enabled mobile phone should see the forecast for a chosen city in the form of texts and/or images. On a screen of the mobile phone, the temperatures and WBMP images symbolizing the weather conditions will be displayed [18]. On the PDA, the user will get colored JPEG pictures and more detailed information about the weather, due to the larger screen. Navigation structures and pagination will be different, but the user will be able to see the weather forecast for the same cities. User experience in the form of messages like “cannot display images” or “deck size is too large” is considered as a delivery of a non-equivalent functionality [73]. The W3C recommends the use of device-independent Web page identifiers and tries to keep the Web unfragmented [73].

The creation of a mobile top-level domain would therefore separate the content displayed on regular computers from the content displayable on mobile devices, and would result in two parallel Webs - the traditional one and a new, mobile Web.

From the author’s perspective, it should be possible to provide a functional presentation in response to a request associated with a specific Web page identifier, in any given delivery context that has an adequate access mechanism (DIP-3) [167]. This principle is simply a restatement of the aforementioned principles from the author’s perspective. The adaptation process should provide a presentation that allows the user to successfully access a Web page and to get information from it or to complete the interaction intended. If a functional presentation of an application cannot be delivered due to inherent limitations in the access mechanism, an appropriate error message should be shown to the user (DIP-4) [18]. Additionally, it should be possible to provide a customized and harmonized presentation depending on device features (DIP-5) [167]. The author should be able to control the presentation on different devices. It is unrealistic to expect from the author that he/she will create different presentation data for each delivery context [167].
Whenever possible, the authored source content should be reused across multiple delivery contexts [18].

Delivery perspective Delivery-related adaptation principles encompass the characteristics of delivery context and delivery preferences [18]. The word “delivery context” suggests a set of attributes that are given for a particular delivery environment [167]. The adaptation software should be able to associate the characteristics of the delivery context with a request for a particular Web page identifier (DIP-6) [167]. Unless the characteristics of the delivery context were made available to the adaptation process, it is not possible to find out whether a specific presentation of content can be delivered in this context, or how to generate a suitable presentation [18]. The user should be able to provide or update any presentation preferences as part of the delivery context (DIP-7) [167]. The delivery context should enable the creation of context-aware applications. If the user provides presentation preferences, they may be used in the adaptation process to offer a more suitable presentation, after taking into account the constraints of the network and the device in question. The process should allow the user to obtain the most appropriate presentation with regard to her/his abilities and all circumstances.

In device-independent approaches, capabilities of devices are the most important part of the delivery context [18]. Relevant features include screen characteristics, supported languages and formats, input/output capabilities, speed of network connections, maximum size of the downloadable content, processor and memory capabilities. Screen capabilities refer to the screen size, supported font types and sizes, as well as to displayable colors. Mobile devices can support various markup languages such as WML, XHTML, HTML, CHTML, HDML, or Java ME and different multimedia types (audio and video formats, various types of images). Input capabilities of mobile devices are different from the QWERTY keyboards, used with traditional computers. Most devices do not offer a full keyboard, but rely on touch-sensitive screens or phone number keyboards with 12-15 buttons. Furthermore, mobile devices can provide only one output type (e.g. textual/graphical) or support different output modalities (e.g. voice, video, text, etc.). The connection speed, acceptable payload size, processor power and memory capabilities are features that can also influence the user experience, and should be considered in the process of application design. Furthermore, a document or application
exceeding the maximal allowed size or free memory cannot be displayed on a device and advanced, resource-intensive Java ME applications should not be provided for devices with limited processor or memory capabilities.

4.6 Adaptation of Content of Web for Mobile Devices
The need of adaptation arises only because of the wide variety of Mobile browsers available on various devices. Our aim in this chapter is to generate architectural view of automatic adaptation process of a web page on mobile device using proxy based adaptation method. To meet this end, we present various process and modes of adaptation, types of adaptation and adaptation tool features.

4.6.1 Process and Modes of Adaptation
Adaptation process is carried out in two simple steps:-
1. To analyze the content according to the specification of devices
2. Transformation of content according to analysis made.

Adaptation can be achieved with any of the following three modes:
1. Designing of Alternative CSS
2. Developing multiple versions of web pages
3. Automatic content adaptation using adaptation engines

However, in current scenario automatic adaptation procedure is the most convenient method to be used. Designing alternative CSS helps in controlling the display of elements and images. By this method, one can detect the device and then the appropriate CSS file can be linked.
The second mode creates multiple versions of the same web page and links the pages to the devices according to their specification.
The third mode, being generally used, converts the content according to device specific version. Its popularity lies in the fact that, it allows creating content in one format only.
In the present study, we focus only on the automatic content adaptation process.
4.6.2 Types of Adaptation

Web adaptation is broadly classified into three broad categories on the basis of the place where adaptation takes place.

1. Client Side
2. Server Side
3. Proxy based / Intermediate adaptation

In Client side adaptation, the required tools and algorithms to convert the content to user needs takes place at client device i.e. Mobile Device? Client side adaptation give flexibility to designers to design the web page only once, as same content is delivered to every device.

In server side adaptation, the adaptation process is completed at server thus reducing the computational time and the adapted page is send to the client through the network. In this mode, several versions of the same web page are stored at the server and the page which meets user need is delivered according to his preferences.

In Proxy based approach, a proxy server analyzes and transcodes the content on-the-fly, before sending the result to the client. [154]

4.6.3 Web Adaptation Tool features

The following are the features of the tools used for web content adaptation:

1. Adaptation for almost each and every device.
2. Automatic content adaptation
3. Filtering of data
4. Should minimize bandwidth requirement.
5. Ease of navigation across pages
6. Automatic site awareness.
7. Automatic fault management and recovery.
8. Optimized access to flash and multimedia content
9. Cost effective
4.6.4 Architectural view of Automatic Adaptation process of a web page on mobile devices using Proxy Based Adaptation Method

In our study, we adopt proxy based adaptation method for the architectural view of automatic adaptation process of a web page. In view of that our architectural view of adaptation process is as given below:

![Figure 4.1 Architectural View of Automatic Adaptation Process](image)

As pointed out in the above figure, adaptation is needed, whenever a mobile device tries to access a web page from the internet. Adaptation is achieved with the help of various free and open source software’s (FOSS) available on web. As we are focusing on developing a mobile web adapter for educational data, security reliability and quality have to be ensured. Keeping in mind all the above factors, we have tried to design a framework for kids’ mobile adapter so that secure, easy and reliable environment can be provided to them.

The adaptation framework consists of the following three components:-

1. Web Adaptation Engines/Server
2. Client profile determining software’s
3. Processors
The working of these components is elaborated with the help of following diagram.

![Diagram](image)

**Figure 4.2 Elaborated view of Automatic Adaptation Process**

The description of the components in the above framework is given below:

**A. Mobile Web Adapter Engines/Servers**

The purposes of these servers are to accept the request from the client device and search that particular page on various web servers. If found, pass that page to processor for adapting the content according to client’s Profile. Most of these engines have built in processors and client profile determination software in it. While the engine passes the request to the web server for searching the requested page, the engine itself uses Client profile determination software to determine the device capabilities. Varieties of these engines are available as free and open source software. A Few of them are:-

- Info Gin’s Intelligent Mobile Platform (For link see [68])
- Byte mobile’s Web Fidelity Suite (For link see [17])
- Novarra’s Vision Server + Native Browser (For link see [107]).

**B. Client Profile determination Software**

As mentioned earlier, adaptation is needed because of the variety of browsers supported by the client devices. So determining the client profile before adaptation is most important step in adaptation process.
The factors that must be taken into consideration are:

1. Device Specification (Memory, Processor, Display statistics)
2. Bandwidth availability
3. Preference of user (Font, Images, Navigation etc)

Device Specification is a process in which all the mobile device related information is captured and stored in a file and the adaptation is done according to the information stored in file. This can be achieved with the help of various Free and open source software’s like

- WURFL [87]
- WALL [88] and
- UAPref [91] etc.

C. Adaptation Processors

The final step requires a processor for selecting, modifying and generating the content so that the device specific result can be formed. Now a days these processors are inbuilt and integrated part of web adaptation engines so separate processors may or may not be required. Examples of such Processors are:

- Cocoon (Apache) [1]
- WEMP (IBM) [69]
- MyMobileWeb (Morfeo) [92].

Other tools available for Images and animation are:

- GAIA Image Transcodes
- PHP Image Rendering Library
- Image Server
- HAWHAW

The framework discussed in this chapter has been integrated into the MBEWCM software in chapter 5 of the present thesis.
4.7 Examples of Content Adaptation Systems

There are various systems which have successfully implemented content adaptation and are working and giving services to the user as well [154]. Few of them are:

1. Oracle Application Server Wireless
2. Sun Java system portal server mobile access
3. Web Sphere Transcoding publisher
4. MobiXtar rich media service center
5. Cocoon and Axkit
6. Morphis wireless content transcoder

Oracle Application Server Wireless

Oracle is better known for its database products. Oracle Application Server Wireless [23][165] is the mobile component of the Oracle Application Server. It includes Multi-Channel Server which abstracts the underlying networks, protocols, devices and gateways to one protocol and one language, namely HTTP and XML. The server enables applications to be accessed through multiple delivery methods and devices such as 2-way pagers, devices with SMS, MMS, voice, or WAP capabilities, and handheld computers. The server automatically translates applications written in Oracle Wireless XML, XHTML Mobile Profile, or XHTML + XForms for any device and network. For example, an XHTML + XForms application passed through the Multi-Channel Server will be translated to VoiceXML if a phone is accessing the application through a voice interface or to WML if a WAP phone issues the request. The tools to support the most popular devices are maintained and regularly updated by Oracle. As new devices come to the market, support is added to the Multi-Channel Server. The framework also supports session management, service linking, and location awareness.

Adaptation services provide device-specific adaptation of images, ring tones, voice grammars, and audio and video streams. For example, images can be dynamically adapted to suit the image format, color depth, size and aspect ratio requested by the device. Ring tone adaptation allows for conversation of ring tone to formats supported by the most popular phones. MMS messages can be authored natively in SMIL or in
XHTML. Messages authored in XHTML are automatically adapted for devices by the product. Adaptation allows a message to be written once and automatically optimized for any target device. By using the Web Clipping part of the product the developer can navigate to the web page containing the desired HTML content, select the portion of the page for delivery to mobile devices. Personalization services enable users to define their preferences and customize their mobile experience. Users can create multiple profiles of their subscribed services. For example, users can define a set of services to be used with a PDA and another to be used through a voice interface.

**Sun Java System Portal Server Mobile Access**

The Mobile Access pack is an extension to the Sun Java System Portal Server. It enables wireless access to the portal and extends the core portal services (session handling, authentication, logging, e-mail, calendar, address book, discussions, etc.) to mobile devices. Java System Portal Server Mobile Access can dynamically render and deliver personalized and aggregated content to users with wireless mobile devices, such as cell phones, PDAs, and smart phones, over any wireless network connection. Sun has developed a set of APIs for processing CC/PP in Java and these APIs are also incorporated in the Mobile Access pack to enable the device independent code. The server dynamically renders content into appropriate markup and tailors it to make best use of the requesting device's features. The Mobile Access pack uses a templating system based on JSP technology for accessing applications or XSL style sheets for accessing XML content. XSL templates are used to transform the XML data into any markup language including XHTML, WML and VoiceXML. [19][89]

**Web Sphere Transcoding Publisher**

Web Sphere Transcoding Publisher [21] is server-based software that dynamically translates Web content and applications into multiple markup languages and optimizes it for delivery to mobile devices based on user preferences and device capabilities. Content adaptation for many devices and languages eliminates the need to maintain multiple versions of a web site. The product can convert images to links to retrieve images, convert tables to lists, and remove comments and features not supported by the device.
(e.g. applets and Shockwave files). The product comes with a wide variety of standard transcoders including:

- HTML to WML, cHTML, HDML, PalmOS HTML, and voiceXML
- XML to wide variety of formats through XSLT style sheets
- JPEG/GIF/WBMP image transcoding and rescaling
- Natural language translation

It is also possible to develop customized transcoders which, for example, handle other image types or Transcodes video and audio clips. IBM has also a product similar to Sun's mobile portal presented earlier called Web Sphere Everyplace Mobile Portal.

**MobiXtar Rich Media Service Center**

The MobiXtar RMSC [42] is a mobile multimedia optimization and delivery platform, specifically designed for mobile messaging systems, portals and download services, which optimizes, adapts and de-livers multimedia content to meet device capabilities and network conditions. It performs multiple multimedia processing functions such as transcoding and optimizing as well as message rendering and delivery. The system enables subscribers to retrieve content through a variety of mobile terminals, each with different display, processor, and multimedia capabilities. The system also enables the composition of MMS messages, optimization of WAP pages and delivery of MMS, email and streaming content. The product converts multimedia content (still images, animations, audio and video) in real-time. For example, it can make conversion between WMT, MPEG4, H.263, AVI and MPEG1 video formats.

**Cocoon and Axkit**

Cocoon [76] is an open-source Java server framework that allows dynamic multi-channel Web publishing of XML content using XSLT transformations. By relying on XML to describe content, and XSLT as a means of transforming that content into multiple formats, Cocoon provides a platform for building applications with separation between content, logic, and presentation. Cocoon can serve static files as well as dynamically generated responses with wide variety of output formats including XML, XHTML, PNG,
JPEG, SVG, and PDF. Cocoon includes CC/PP and UAProf support through another open-source product called DELI, which is a library that provides an API to allow Java servlets to resolve HTTP requests containing delivery context information from CC/PP or UAProf capable devices and query the resolved profile.

AxKit [145] is a product similar to Cocoon but built in Perl. It provides on-the-fly conversion from XML to any format. To our knowledge AxKit does not have support for CC/PP or UAProf, but it can deliver data to different devices using the same set of URLs for all media types (screen, handheld, tv, etc.). AxKit supports GZipped output encoding which is valuable for slower networks. Users of Axkit include Opera Software.

**Morphis Wireless Content Transcoder**

Morphis [63] is an open source Java based Web content transcoding, transformation, translation and aggregation framework. It was primarily built for retrieving and translating XML based documents, but it is also capable of performing any type of translation: binary, plain text, or text markup. Therefore, Morphis is able to convert XML into HTML or WML, while also being able to scale, crop and convert images on the fly. Central to the Morphis solution is WAX (Wireless Abstract XML), a set of tools and an abstract markup language used to author content for wireless applications. Content is written once in WAX language, and then translated to various wireless languages via XSL style sheets.

**Conclusion**

Since the web page adaptation is a must for providing the retrieved web content via the mobile interface. We have to put forward in this chapter, the important details on web content adaptation with an architectural view of automatic adaptation process of web pages on mobile device using proxy based adaptation method. The designed framework includes mobile web adapter engine with client profile determination and adapting modules. The framework discussed in this chapter has been integrated into the MBEWCM software in chapter 5 of the present thesis.