Design of an e-Learning Content Visualization Module

Evgeniya Georgieva
University of Rousse
8 Studentska Str, 7017 Rousse, Bulgaria
+359 82 888 276
egeorgieva@ecs.ru.acad.bg

Tsvetan Hristov
University of Rousse
8 Studentska Str, 7017 Rousse, Bulgaria
+359 82 888 276
thristov@ecs.ru.acad.bg

ABSTRACT

In this paper is described a design of an e-Learning content visualization module. The existing methods for mobile device recognition are examined. Content adaptation approaches – server-side, proxy-side and client-side are also considered. Conclusions for appropriate technologies which to be used are made.

Categories and Subject Descriptors

D.2.2 [Design Tools and Techniques] Flow charts, Modules and interfaces, H.5.2 [User Interfaces] Graphical user interfaces (GUI), Screen design, I.7.2 [Document Preparation] Hypertext/hypermedia, Markup languages, Multi/mixed media, K.3.1 [Computer Uses in Education] Distance learning

General Terms

Design, Management, Languages.

Keywords

e-Learning, m-Learning, Content management, Device recognition, Content transformation.

1. INTRODUCTION

The fast grow of ICT leads to the appearance of wide variety of devices which the people use in their daily round – desktop computers, Notebook computers, Tablet PC, Personal Digital Assistants (PDAs), cell phones and smart phones. One of their common characteristics is that these devices provide access to information via Internet. At the same time they significantly differ from each other in the following characteristics: web browser they use, supported markup languages (HTML, XHTML, cHTML, WML, etc.), supported script languages (JavaScript, JScript, VBScript), file formats and the screen resolution. These different parameters of the devices lead to the necessity for applying an adaptation method to correct visualization of already developed electronic content.

The solution of the task for correct content visualization on different devices is very important for e-Learning. On the one hand it will ensure learners' ability to access education content anytime and anywhere. On the other hand once more development of education content especially for mobile devices will be avoided.

In the University of Rousse, Bulgaria an e-Learning software platform named e-Learning Shell is developed. It enables lecturers to publish their own WEB-based course without having any skills and knowledge in HTML, JavaScript and other WEB-programming languages or tools. The user friendly interface and simple mechanisms for uploading learning content allows building a variety of courses in various knowledge domains. There are 80 courses, uploaded on the university server with 3800 registered users.

Currently existing courses in the e-Learning Shell (eLSe) Software Platform are available only from desktop computers. They cannot be used with mobile devices.

For this goal a new content visualization subsystem have to be designed and developed. On the one hand this subsystem has to deliver content to the users that use PCs with browsers, and on the other hand it has to deliver e-Learning content to users that use mobile devices with specific browsers.

2. LAYOUT

2.1 Mobile Device Recognition Methods

The application of device recognition methods is an important step before the process of content adaptation. Currently, servers and proxies can determine the identity of a particular device using the request header field in the HTTP protocol. In addition there are three alternative methods: the W3C composite capability / preferences profile (CC/PP), the WAP User Agent Profile (UAPROF) standard and Wireless Universal Resource File (WURFL).

- HTTP User-Agent Header

The web browsers and servers use the HTTP protocol to transfer information on the WWW. It includes mechanism for content presentation which browsers can accept. The server decides what kind of information to send depending on the device profile. Each HTTP request includes Accept Header [7], which indicates the types of data, which the browser can accept. In addition to the Accept Header the client sends User-Agent Header [7]. It identifies the client device and contains information about the browser, operation system and sometimes hardware information.

As the number and the kind of devices, which have internet connection, constantly grow up, the needs of content designed to different devices also grows up. That's why the information from User-Agent Header is not sufficient.

- Composite Capabilities / Preferences Profiles (CC/PP)

The specification Composite Capabilities/Preferences Profiles (CC/PP) from World Wide Web Consortium [4] documents standard way, which allows devices to transmit their configuration details and abilities (screen resolution, audio characteristics, frequency band) to web servers. CC/PP

specification provides universal profile that describes the devices' characteristics. CC/PP is projected to be independent. The connected specifications as UAPROF, unlike CC/PP, define a variety of dictionaries describing the devices' characteristics.

- WAP User Agent Profile (UAPROF)

Another way to identify the user device profile is using the User Agent Profile [9] specification. It is a specific CC/PP dictionary describing mobile devices and defining an effective way for CC/PP content transition via wireless nets. Mobile phones conformed to UAPROF specification provide CC/PP description of their characteristics on the server. Content servers, gateways and proxy servers can use this information and optimize the content for the device and the consumer. Information is in XML format. When mobile device sends request to server, it also sends an URL address to its mobile profile. This is carried out by adding of X-Wap-Profile Header to request. This header indicates the server where to find the device profile. The content server extracts the necessary information for the client from device profile repository and can store it, so that it can be used later. WAP gateway or HTTP proxy must support working of UAPROF header.

- Wireless Universal Resource File (WURFL)

The mobile device profile can be identified using the open source project Wireless Universal Resource File (WURFL) [8]. It is a configuration file containing information about the features of mobile devices offered on the market. The main goal of the developers of this file is to support maximum information for existing wireless devices that have an access to WAP pages. In this way the developers have an opportunity to develop better applications and client services. This is an open source project and is designed to operate in WAP environment. WURFL is a universal database containing information about more than 1500 devices and over 300 profiles describing each device. One of the requirements is that this database should be accessible on all platforms - that is why the selected language is XML. WURFL project has some advantages compared to UAPROF: 1) WURFL file can be stored on the server and it is not necessary to be accessed remotely; 2) Each device characteristics can be shaped.

2.2 Content Adaptation Approaches

The content adaptation is a process of selecting, generating or modification of content (text, images, audio, video), so that it can be presented by devices with different functional possibilities. In case of access to website using desktop computer, on its screen can be seen the original web page (with no modifications). If the same website is accessed via mobile device, images must be resized and compressed, text must be formatted in one column, and video must be presented as a text description or as an image depending on available frequency band. Content adaptation can be done in one of the following ways: by web-server, by medial proxy server, by client (Fig.1).

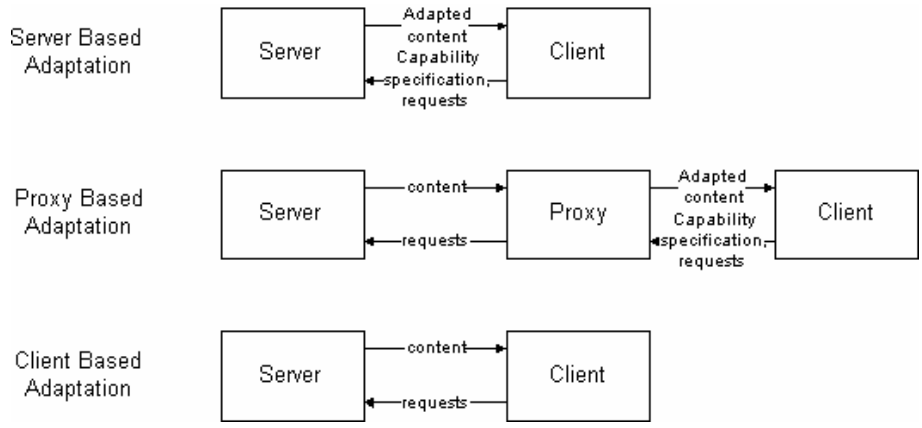

Fig.1. Content adaptation approaches [1]

- Server-based approach

In server-based approach the web server functionality is extended with modules for content adaptation. Traditionally, a great number of versions of one and the same content are stored on the server, but only the content, which coincides with the client's profile, is presented. The content delivery time is reduced to the delivery time of the already adapted content. A more common approach to content delivery for different device types is to store the information in XML format, then to use XSLT for its transforming in the preferable markup language. Examples for XML+XSLT solutions are Cocoon and AxKit. Cocoon [6] is an open code Java server platform. The content is stored in XML format and via XSLT is transformed in different formats. Cocoon provides platform for applications (content, logic and presentation) working out. AxKit [2] is a product similar to Cocoon, developed using Perl. It supports transformation from XML to every format – HTML, WML, text, etc.

- Proxy-based approach

In proxy-based approach a proxy server analyses and transforms the content before sending it to the client. Proxy server also caches the adapted content, which can be used later. In this way the transformation is done only once. In order to be able to send the right content, the proxy server and the web server must "know" what type of device makes the request. An example of proxy based approach is AvantGo [3] which is a free service that delivers thousands of mobile websites to consumers on their handheld personal digital assistants (PDAs) and smart phones.

- Client-based approach

In client-based approach the necessary transformation is done by a consumer's device. In this approach the author of content is not supposed to work out different variants of content. For example, in its mobile browsers Opera Software uses client-based adaptation technology, called Small-Screen Rendering (SSR) [5]. This technology transforms the content in a way that it is shown in a full screen, eliminating the need of horizontal scroller. Another way for client adaptation is using of (X)HTML and different CSS for each type of device.

2.3 Visualization Module Design

Taking into consideration the above mentioned adaptation approaches, the server-based approach will be used for the needs of the visualization module. HTTP User-Agent Header and Wireless Universal Resource File will be used for recognizing the profile of the client device.

Taking into consideration the functionality diagram on Fig. 2 the visualization subsystem can be divided into the following modules:

- Content visualization module

This module processes the visualization request from the core of the system and retrieves the resources IDs, types and their order from the database.

- Resource visualization modules

The main function of these modules is retrieving the resource data from the database and processing the visualization templates. As it was mentioned above, each resource type requires different visualization approach depending on its data and structure. In this order ten resource visualization modules have to be developed for each type. If a resource cannot be visualized on a device, the user will receive the metadata of the object.

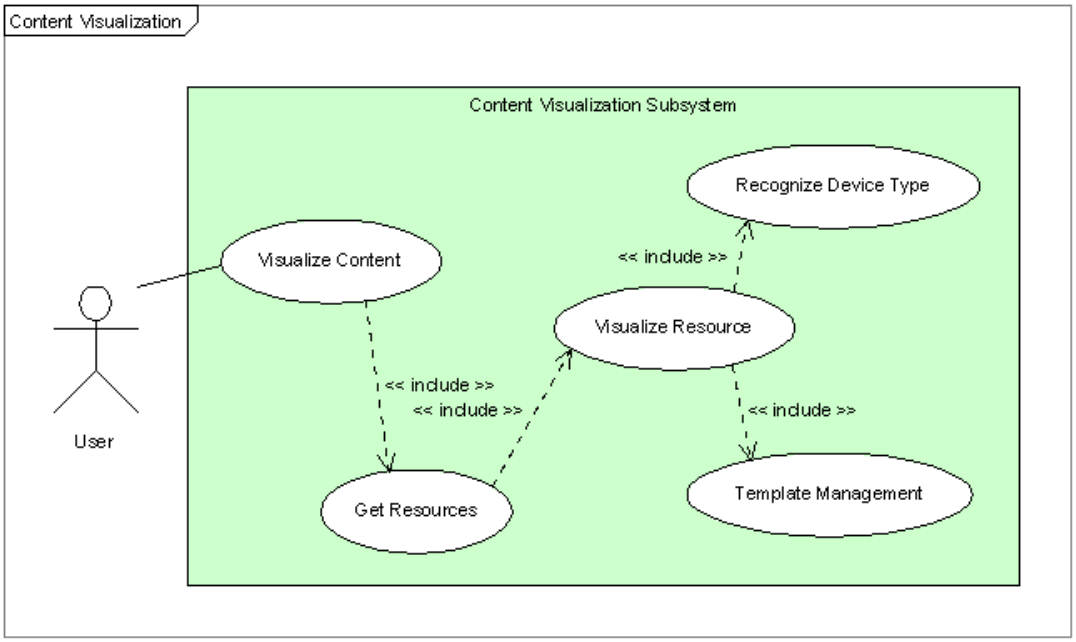

Fig.2. Use Case diagram of Content Visualization Subsystem

The learning content of the WEB-based courses developed using the authoring tool of the e-Learning Shell software platform consists of resources. Depending on their content, these resources can be divided into the following types: text, file, multimedia, tests, message board, Frequent Asked Questions (FAQ), timetable, forum, workshop and chat. The visualization of each type requires a different approach and a separate set of templates for each device type. That is why in the process of development there will be developed separate modules with equal structure for visualization of every resource type.

The functionality of the visualization module is shown on Fig. 2. The learning content visualization requires recognizing the type of separate resources and visualization of each one. This process includes device recognition, selecting a proper template depending on its profile and working of the results.

- Parser

While the visualization system is based on templates stored into the database or the file system, a specialized parser module is needed for the template processing. This module retrieves the visualization templates, processes them and returns the result to the user's device.

- Device Recognition

This module retrieves the user device information from the HTTP header of the requests. On this basis it creates the device profile using WURFL for the mobile phones and HTTP headers for the PDAs and saves it into the server session. The most important information in this profile is the supported markup language (HTML, XHTML, WML, etc.) and the screen resolution.

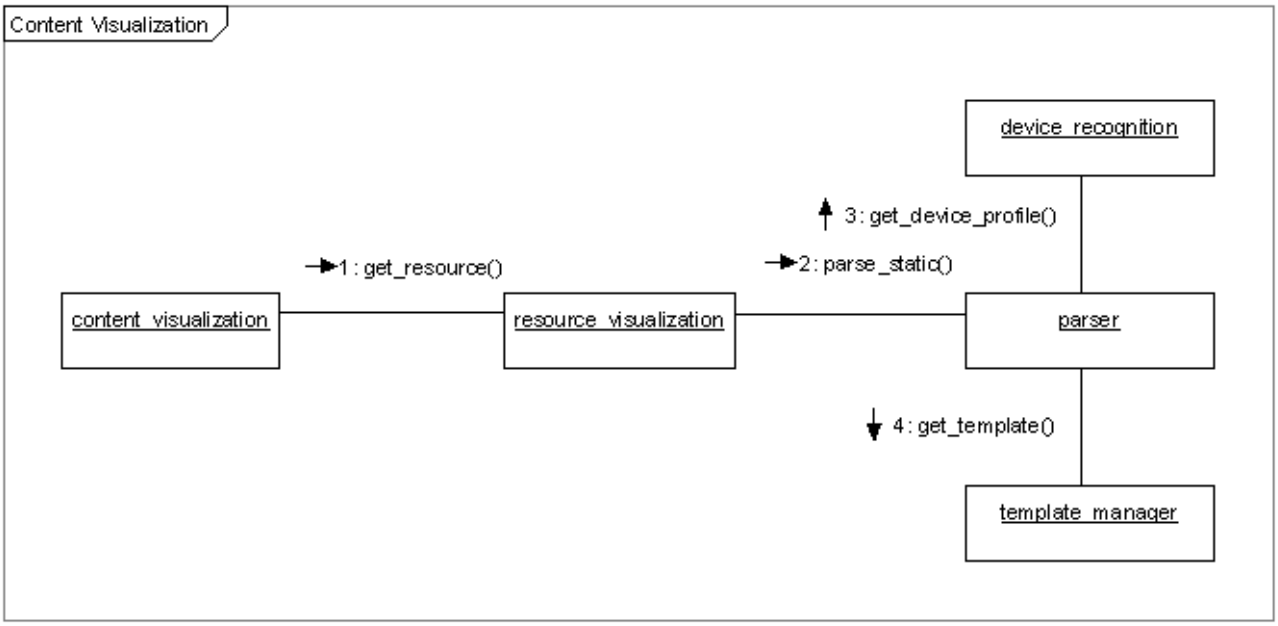

Fig.3. Communication Diagram

- Template manager

This manager retrieves the template file physical name and location by using the IDs stored into the database, taking into consideration the device profile and the requested template type.

On Fig. 3 the communication diagram is shown. After retrieving the resource type the content_visualization class calls the get_resource() member of resource_visualization (1). Resource_visualiation in this case is a common name for all the resource visualization classes. They all have common structure and performance. Resource_visualization calls the template manager method parse_static() from the parser class (2). This class gets the device profile by calling the get_device_profile() function of the device_recognition (3). On the basis of the device profile the parser class retrieves the appropriate template location by calling get_template() method of template_manager (4), parses the template and returns the result content to resource_visualization class. After receiving the visualization information for all resources in the current view the resource_visualization class generates the visualization code, applies the design template and returns the result to the user.

3. CONCLUSION

This paper describes a module for e-learning content visualization. The module uses XML template based server-side adaptation. It recognizes the user device type analyzing the HTTP User-Agent Header and using WURFL file. On this basis it is able to adapt the learning content depending on the user's device abilities, the user rights and the type of content. Preliminary tests show that the module is fully functional and can be used successfully with e-Learning Shell (Learning Content Management System), developed in the University of Rousse.

4. REFERENCES

[1] Butler, M., Current Technologies For Device Independence, http://www.hpl.hp.com/personal/marbut/ currTechDevInd.htm.

[2] Apache AxKit, http://axkit.org/.

[3] AvantGo, http://www.avantgo.com/frontdoor/index_uk.html.

[4] Composite Capability/Preference Profiles, http://www.w3.org/TR/CCPP-struct-vocab/.

[5] Opera's Small-Screen Rendering, http://www.opera.com/products/mobile/ smallscreen/.

[6] The Apache Cocoon Project, http://cocoon.apache.org/.

[7] Tutorial about Detecting User Agent Types and Client Device Capabilities, http://www.developershome.com/wap/detection/.

[8] Wireless Universal Resource File (WURFL), http://wurfl.sourceforge.net/.

[9] User Agent Profile Approved Version 2.0, www.openmobilealliance.org/release_program/docs/UAProf/ V2_0-20060206-A/OMA-TS-UAProf-V2_0-20060206-A.pdf.