Models and Methodologies for Automated Creating of Webpage Mobile Versions

Todor Rachovski 1, Emil Hadzhikolev 1, Stanka Hadzhikoleva 1

1 University of Plovdiv “Paisii Hilendarski”, Department of Mathematics and Informatics, 236, Bulgaria Blvd. Plovdiv, Bulgaria

e-mail: todormr@gmail.com, hadjikolev@uni-plovdiv.bg, stankah@uni-plovdiv.bg

Abstract. There are still many outdated web pages on the Internet that are difficult to browse with small screen mobile devices. Software tools are needed to aid mobile users by generating mobile versions of web pages, depending on the parameters of the specific mobile device, which is used. The main aim of this paper is to present a technique, a conceptual and computer model of an application for automated generation of mobile pages. In the following are considered methodologies for redesign and dynamic design of web pages. A five-layer architecture of the Mobiles Online application is presented. It includes database layer, database management layer, business logic layer, web services layer and presentation layer.

1. Introduction

In recent years, there has been a tendency of increased access to websites via mobile devices. There are various reasons for that - finding information, shopping, training and other business activities. According to an independent web analytics firm StatCounter, in October 2016, the internet traffic, generated from mobile devices and tablets, exceeds the one, generated from desktop computers [1].

The use of mobile devices to access websites sets new high requirements for website developers. The efforts of the developers are redirected from creating websites with artistic and attractive design to providing relatively simple design, that looks good on different mobile devices. Responsive web design is the design that provides optimal web visualization easy-to-interact pages that are suitable for various mobile devices with different screen resolutions - laptops, tablets, smartphones, etc. The reason for this adaptation is to minimize the need to resize and to scroll the information on the screen. The improving of HTML and CSS standards (respectively to set the structure of webpages) and their compliance with modern browsers have made the job of the website developers easier. Their efforts have changed from maintaining websites for different browsers to creating visually appealing websites. The main issue nowadays is to create websites that are visualized fine both on desktop computers, laptops, tablets and smartphones with different screen resolutions. The solution for that is creating websites by using a “resizable” design [2], [3].

2. Adaptive and responsive design

Many people do not make a difference between the following terms “responsive design” and “adaptive design”, and use them as synonyms. The specificity of the two terms is related to the different approaches, used to design webpages suitable for visualization on both small and large screens.

The term “responsive web design”, introduced by Ethan Marcotte [4], refers to the websites that are displayed equally well on different devices like mobile phones, laptops, tablets etc., without the need of changing the source code. Although the term “responsive web design” gained popularity relatively
recently (by 2012), to date, designing and building new websites, that do not use this technology are considered unprofessional.

The term "**adaptive web design**" is used in the case of websites that need to look good on devices that use a specific screen resolution [5].

The common between the two terms is that both responsive design and adaptive design are related to creating webpages that are optimized for screen size or type of device that is used for reviewing them. The main task in both cases is related to creating webpages suitable for the visualization of screens of varying sizes. Most often it is required that the website must look equally good on at least 3 different devices like a small mobile phone, a tablet or a large desktop monitor.

**Advantages of using adaptive design** are:

- better accessibility from different mobile devices and desktop computers;
- the design looks like the main design, but it is optimized for devices with different screen resolutions;
- better indexing by Google (SEO);
- there is a need for creating only one website that looks equally good on different devices;
- the long-term solution, from the point of view of the evolving and diverse market for mobile devices;
- good looking design with easy navigation.

The main **disadvantages of using responsive design** are the following:

- loading speed - may be a problem with large sized websites and websites with lots of videos and images;
- high budget - to implement responsive design a full rewrite of the code is required, which is usually expensive.

There are a lot of tools in the Internet aiming to analyze and assess the suitability of a web page for mobile devices - WompMobile [6], mobiReady [7], Mobile [8], PageSpeed Insights [9], Bing [10], etc. One of the most famous and free tools for webpage analysis is Mobile-Friendly Test Tool [11], developed by Google. It checks if a web page has a responsive design, highlights the weaknesses of the design and specifies resources with information about optimizing it. The main features that the tool tests are the following:

- Using Flash. Most mobile browsers do not support Flash. The content, based on Flash is not visualized, so the use of such is not recommended;
- Configure a view (Viewport). Each page must have a configured view (via HTML meta tag viewport). The view points to the browsers on how to adjust their dimension scale for the web page so that the design adapts according to the device on which it is being visualized.
- Fixed webpage size. Some pages have a fixed width that is contrary to the requirements for responsive design;
- Small text size that makes reading more difficult;
- Placement of controls and buttons. The placement of controls and buttons near to each other can lead to miss clicks and to make them harder to use.

3. Models and methodologies for automated creating of web page mobile version

3.1. **Basic requirements**

In the context of the constantly changing web standards imposed by both, users and search engines, the following basic requirements are set, for the proposed technology:

- To maintain responsive design for different types of mobile devices with different screen resolutions, and for different browsers;
- To allow adding new themes for new mobile devices including ones with different size and resolutions than the standard ones;
- To have an intuitive user experience interface;
- To store the built-in mobile versions of the web pages so if the website is refreshed it will not require to be converted again;
- Collecting error information in the process of converting so that they can be removed in the future;
• To be open for future extensions and changes, its components can be used repeatedly, replaced or used by other applications etc.

3.2. Architecture
The architecture is multilayered and includes: Database, layers for Database Management, Business Logic, Web Services and Presentation Layer (fig. 1).

All data is stored in Database: app users, generated mobile pages, error messages and some statistics. For greater flexibility, scalability and the ability to record data without a predefined scheme, it is appropriate to use NoSQL databases. This guarantees application stability and fast scalability of the volume of stored data.

The Database Mmanagement Layer is based on the concept of ORM technology (Object-relational mapping) to convert data by using object-oriented data programming languages. It contains information about the access of the application, presentation of database information in the form of classes and objects, etc. This layer creates a “virtual” object database that can be easily manipulated through the means of a specific programming language without the need for additional processing.

The Business Logic Layer contains the components that perform the actual work on building the mobile page. The main component in it is the HTML parser, who performs an analysis of the original web page, creates a tree object model, optimizes web page objects, and builds an interim, non-stylized version of the web page.

The Web Services Layer gets processed information from the Business Logic Layer, converts it into JSON format and transmits it to the Presentation Layer. This layer defines interfaces for accessing the web services and hides the details of the business process of building the JSON file. It allows third-party access to JSON files, respectively for adding styles developed by other suppliers.

The Presentation Layer is responsible for choosing and using the appropriate CSS style for the mobile page, building depending on the mobile device that is being used. Additional information about the conceptual model is presented in [12].

The advantage of using this architecture is that it maintains a high level of abstraction and it remains open, easily expandable and flexible.

4. Software prototype Mobiles Online
The aim here is to present the development of a software prototype, as well as to explain the basic requirements and steps in building a mobile web page. The Business Logic Layer as well as the Database Layer are also explained here.

4.1. Basic requirements
Mobiles Online is a tool to aid mobile users by generating mobile versions of web pages, depending on the parameters of the specific mobile device, which is used. It serves as a moderator between the client agent (desktop or mobile web browser) and the real server, where the target web page is hosted (fig. 2.) The user requests a web page by writing its address on Mobiles Online. The request is transferred to the target web server that returns the original HTML page. Mobiles Online processes the received data and generates a mobile version of the page, which is provided to the customer. Mobiles Online works as a proxy server, which changes the visualization way of the desired content, with the knowledge of the customer.
The main steps of the application work process are as follows:

**Step 1**: The user starts in the browser the web application Mobiles Online.

**Step 2**: The user enters the URL address of the web page in the box “Search”.

**Step 3**: The indicated address is being validated and if it is invalid, the customer is given an error message.

**Step 4**: The content of the required resource is being pulled out. If the content is an invalid HTML document, the customer is given an error message.

**Step 5**: The HTML parser analyses and processes the information and builds a hierarchical object model of the web page elements.

**Step 6**: A model of the document, which is suitable for mobile styling, is being generated.

**Step 7**: On the created model of the web page is applied a style, which is consistent with the parameters of the called mobile device.

**Step 8**: The generated mobile version is returned to the user’s browser.

4.2. **Business Logic Layer**

The main function of Business Logic Layer, which is the core module, is to retrieve web page information and convert it from ordinary to mobile. The created analyst serves to process the entire HTML document and after processing submits to various functions a part of this HTML, which processes and converts it accordingly.

The steps taken during the overall document processing are the following:

- invoking the HTML document analyzer;
- extracting information from meta tags;
- retrieving the navigation menu;
- retrieving the footer;
- retrieving the title;
- retrieving the basic web page information (the body part).

A XML/HTML analyzer (parser) can be used to retrieve information from a web document. There is one in almost every programming language. Between the two standards - XML and HTML there are significant differences, which creates the need for analysis and individual solutions. It is important to note that XML documents must have a valid structure - for example, each opening tag must have a properly placed closing tag. In contrast, HTML pages may not be described correctly. In the case of an incorrect HTML document (lack of closing tags, incorrectly nested items, incorrectly tagged elements, etc.), browsers try to build the correct content visualizing it without informing users about the errors that occurred. Because of this tolerance, many of the web pages, created years ago, do not contain valid HTML. Incorrect structuring of an HTML document may cause problems while it is being parsed.

Some of the most important HTML elements handled by the app are listed in table 1.

| HTML tag | Description |
|----------|-------------|
| <header> | Contains all the header information on the page. Typically, the headline and meta data on the page are displayed. |
| <footer> | Contains information about the bottom of the page. |
| <title> | Page title. |
| <meta> | Meta data, including author, keywords, and more. |
| <span> | Used to group items into line. |
| <div> | Used as an element separator. |
| <input> | Entry field. |
| <form> | Form. |
| <img> | Description of images. |
| <nav> | Navigation menu. |
| <a> | Hyperlinks (links). |

Generating a DOM model is done recursively and consists of a number of hierarchically organized DOM nodes. For example, pulling out the body of the HTML document, starts the searching of the
content between open and closed body tag, which is the first level object. From the content of the body all elements from the next second level are being retrieved.

DOM Node objects have the following main fields:

- Name – current HTML tag;
- Attributes – a collection of attributes contained in the current HTML tag;
- Inner – the content between the start and end tags;
- EndSlash – the variable that indicates whether the HTML tag ends with “/”.

4.3. Database Management Layer

The Database Management Layer links to the Database where the already processed information is stored. For this purpose, classes are created with the tables of the base using:

- Content – here is the link between the MobileContent and RealContent tables;
- MobileContent – the already processed information is stored;
- RealContent – stored initial information;
- RequestError – errors that occurred (for example wrong response to site request);
- Role – user roles;
- Users – user’s data;
- Users_Roles – the relationship between individual users and their roles.

4.4. Presentation Layer

Web API Controller can be found in this project, which is the service that connects the visualization (View to MVC) and the DAL project, also the controllers, where the session of the information exchange, between the views, is created.

In the Web API Controller are several functions to find. The most important between them is the GetData, to which one URL is submitted. The URL could be submitted in a different format or represented by the following expression

```
^(http|https)://([w+]?\w+)\+[a-zA-Z0-9\-\@\#\$\%\^\&\*\(\)\_\+=\?\:\;\']\*)?
```

Examples:

- uni-plovdiv.com
- www.uni-plovdiv.com
- https://uni-plovdiv.com
- https://www.uni-plovdiv.com
- http://uni-plovdiv.com
- http://www.uni-plovdiv.com
- www.Asddsadasd.sdf – wrong, an error message is displayed.

5. Methodology for building mobile webpages

There are two main methodologies (dynamic methodology and redesigned methodology) depending on whether a dynamic generation of the content of the mobile page is performed or a static site is used.

5.1. Methodology for creating mobile apps with reprojection

As is already clear from the application model, and following the architecture of figure 1, in a very simple way, using a described algorithm, a user can create their own page for an existing application. There are a few basic things to consider: building an adaptive design, configuring its work with an application to automatically generate mobile pages, building an adaptive design from existing information. The methodology is structured in the following steps:

**Step 1.** Build an adaptive design - here can be used a number of standards for mobile page visualization and design. The user can design their own view according to the test and depending on the application.

**Step 2.** Configuring the design with an application to automatically generate mobile pages - the necessary thing on the user’s side is a mechanism for accessing the system. This is done by WEB API
and the following functions: Function to extraction and process all information - passing two parameters for address and encoding type. The encoding type is not mandatory.

- All information preview feature - here it is recommended to use it in the administrative part of the application. It returns a list of all processed pages;
- The Page Preview function depending on the page name - as well as the previous function - is recommended here for use in the administrative part. The basic functionality is to provide information about a generated page;
- Function to use the already generated content - typically submitted by the client;
- Mobile page extraction feature - provides a page name parameter and returns all of its content. The place of the parameter is filled in the name of the static page.

5.2. Methodology for creating mobile applications from dynamic pages
The main goal here is to create a methodology for dynamic mobile pages the same as in the reprojection methodology. The basic steps here are:

**Step 1.** Building an adaptive design - the main difference comes out with the building of the adaptive design. In contrast to static, this should be strictly appropriate and simplified, because of its use for different web pages. It is recommended to use all the good practices in pages;

**Step 2.** Configuring the design work with an application for auto-generating of mobile pages. With using of dynamic mobile applications, it is necessary to pull out and generate the newly inserted page every single time. Before that, the name of the new page must be passed dynamically to the system.

### 6. Experiments

6.1. Creating a web page with the methodology for creating mobile apps with reprojection
The first experiment of the presented model and methodologies is related to creating and generating a mobile page. The experiment uses the methodology for creating mobile applications with reprojection and is attached to the website of the Plovdiv University "Paisii Hilendarski": http://uni-plovdiv.bg.

**Step 1.** Build an adaptive design.

When designing is created, it is very important to keep in mind and see contemporary mobile design requirements. There are many software frameworks that are user-friendly and offer ready-made systems with easy-to-use page layout functionality. Some of the most commonly used are Tweeter Bootstrap, jQuery mobile and others.

Creating an adaptive design also requires recognizing the screen resolution of the mobile device. A key point here is the definition of several “media screens” or “media queries” in line with these requirements. The following screens are defined for the experiment:

- @media (max-width:1920px) - consistent with maximum screen devices 1920 pixels
- @media (max-width:1600px) - consistent with maximum screen devices 1600 pixels
- @media (max-width:1440px) - consistent with maximum screen devices 1440 pixels

Figure 3. Plovdiv University web page.
Step 2. Configure your application for working with a mobile model for generating mobile pages. The JavaScript programming language is used to implement this step. It's a scripting language that handles client-side data, specifically in the user's web browser.

Step 3. Create an adaptive design from existing data - like other cases, the information is saved in certain elements.

6.2. Creating a web page with the methodology for creating mobile applications from dynamic pages
The construction with dynamic methodology is almost identical to that with the redesigning methodology. The difference here is that the user can dynamically enter the name of the URL. As functionality can be set various types of templates which are suitable for the current visualization (fig. 4).

Step 1. Building an Adaptive Design - in this case, are created three basic designs (templates):
• Main design - designed on the basis of modern standards for building mobile websites;
• Bootstrap design - designed based on bootstrap technology;
• Specific design - tailored to the specifics of old web standards used (older pages).

Step 2. Configuring the design work with an application for auto-generating of mobile pages. After entering the URL and the processing, the main DOM elements are visualized. With its first loading, the main design is used by default. With the next loading, it is verified, which is the final used style by the user, followed by the JavaScript function for filling the processed page information.

Step 3. The building of adaptive design from the already existing information. At the expense of the previous experiment here must be considered the visualization of the different web pages. Because of the different character of the content, they are divided into three basic templates.

Main design - it is divided into four basic sections:
• Navigation menu section;
• Basic information section;
• Additional information section;
• Information section in the footer.

Additional information about the Mobiles Online prototype is presented in [13].
7. Conclusion
Adapting and creating web sites in the Mobile-era have requirements for responsive design and is one of the major challenges for web developers. In general, the task can be solved in two ways: completely rewriting the website to create a responsive design or to create a new mobile version. A universal solution for users sites is the use of apps that automatically generate a mobile version of a web page and that is relevant to the specific user device. This paper present a technique, a conceptual and computer model of an application for automated generation of mobile pages.

The presented technology is suitable for creating plugins that can be integrated into mobile browsers. This can automatically launch the process of building and viewing a mobile version of a web page and hide the application’s work from users.

References
[1] StatCounter 2018 Desktop vs Mobile vs Tablet Market Share Worldwide (Aug 2016 - Aug 2017) online: http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide last visited 30.10.2020
[2] Kim B 2013 Chapter 4: Responsive web design, discoverability, and mobile challenge in The Library Mobile Experience: Practices and User Expectations ALA TechSource pp 29-39
[3] Mohorovičić S 2013 Implementing responsive web design for enhanced web presence 36th International Convention on Information & Communication Technology Electronics & Microelectronics (MIPRO) (IEEE) pp 1206-1210
[4] Marcotte E Ethan Marcotte Blog online: https://ethanmarcotte.com last visited 30.10.2020
[5] Berry RWD Is Not AWD, What Is the Difference Between Responsive and Adaptive Design? Mockplus blog online: https://www.mockplus.com/blog/post/difference-between-responsive-and-adaptive last visited 30.10.2020
[6] WompMobile online: http://www.wompmobile.com last visited 30.10.2020
[7] mobiReady online: https://ready.mobi/ last visited 30.10.2020
[8] Mobile SEO online: https://varvy.com/mobile/ last visited 30.10.2020
[9] Google PageSpeed Insights online: https://developers.google.com/speed/pagespeed/insights/ last visited 30.10.2020
[10] Bing Mobile Friendliness Test Tool online: https://www.bing.com/webmaster/tools/mobile-friendliness last visited 30.10.2020
[11] Google Mobile-Friendly Test Tool online: https://search.google.com/test/mobile-friendly last visited 30.10.2020
[12] Rachovski T Hadzhikoleva S and Hadzhikolev E 2017 Conceptual Model of an Application for Automated Generation of Webpage Mobile Versions TEM Journal 6 (4) pp 877-882
[13] Rachovski T Hadzhikolev E and Hadzhikoleva S 2017 Software application for generating mobile versions of webpages, Scientific Session “Days of Science 2017” of the Union of Scientists in Bulgaria – Plovdiv (Plovdiv, Union of Scientists in Bulgaria) pp 149-156