

AUTOMATIC EVALUATION OF MOBILE WEB PAGES

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Ankita Sehgal

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

April 2010

Fargo, North Dakota

North Dakota State University
Graduate School

Title

AUTOMATIC EVALUATION

OF MOBILE WEB PAGES

By

ANKITA SEHGAL

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

North Dakota State University Libraries Addendum

To protect the privacy of individuals associated with the document, signatures have been removed from the digital version of this document.

ABSTRACT

Sehgal, Ankita, M.S., Department of Computer Science, College of Science and Mathematics, North Dakota State University, April 2010. Automatic Evaluation of Mobile Web Pages. Major Professor: Dr. Jun Kong.

The use of mobile devices to access the information on the World Wide Web is growing rapidly. The growth has created a surge in the number of webpages being designed for mobile devices. In order to support efficient mobile browsing, some markup languages (e.g., WML and XHTML Mobile Profile) and standards (Mobile Web Best Practices) have been developed to specifically support the rendering of information on mobile devices. Evaluating the mobile webpages manually against the guidelines requires a tremendous amount of work. This research paper focuses on automatic evaluation of mobile webpages, a process which offers many advantages in terms of saving time, reducing costs and many others.

In this paper, an extensive inspection of the 46 most popular mobile websites, according to Mobile Web Best Practices using MobileOK checker [WMC09], has been conducted. The results have been provided and analyzed to make specific and general suggestions to improve the design and development of mobile websites. We believe that the analysis will help developers produce more user-friendly mobile websites while considering the Mobile Web Best Practices. Furthermore, the paper also presents a generic approach to represent the W3C Mobile Web Guidelines as an XML schema, which provides flexibility to design rule schemas in an open format; the schemas can be added to an automated tool that is used to evaluate the mobile webpages. The XML schema can also be used to redesign the rule if there is a change in the guidelines.

ACKNOWLEDGEMENTS

This paper is one of the most significant achievements of my life, and it would not be possible without the love and support of people who believed in me. I express my sincere gratitude to Dr. Jun Kong, whose guidance, advice and support have helped me achieve my academic goal. Without his continuous support, help and direction, it would not be possible to finish this paper.

I would like to thank the faculty and staff of the Computer Science Department for their help and guidance throughout my master's program. I would like to thank my husband Anoop Kohli for his constant support and words of encouragement that helped me to have a positive mindset while working on my master's paper work.

I would also like to thank all of my friends, Anoop Kheerwal, Divya Sehgal, Shaifali, Harvinder, Faraz, Thilak, Vandana, Ashish and Dhruv, for their love and support which helped me feel at home in Fargo.

My sincerest gratitude to my mother, father, sister, brother and cousins who provided me with wonderful memories, support and encouragement throughout my education for which I will always be indebted and would like to pay off by achieving great success in my career.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF TABLES.....	viii
LIST OF FIGURES	ix
CHAPTER 1. INTRODUCTION.....	1
CHAPTER 2. RELATED WORK.....	5
CHAPTER 3. EVALUATION BASED ON MOBILE OK CHECKER	12
Navigation and Links.....	12
Auto Refresh.....	13
External Resources	13
Pop-ups	14
Redirection.....	14
Image Maps.....	14
Page Layout and Content.....	18
Page Size Limit.....	18
Graphics for Spacing	18
Page Definition	20
Page Title	20
Minimize.....	21
Content Format and Support.....	21
Character Encoding Use and Support.....	22
Caching	22

Valid Mark Up	23
Non-Text Alternative	24
Objects or Script	24
Image Size.....	28
Image Resizing	28
Images Specify Size.....	29
Tables.....	31
Table Alternatives, Table Layout and Table Nested	31
Style Sheets.....	34
Style Sheet Support and Use.....	34
User Input	36
Provide Defaults	36
Default Input Mode.....	36
CHAPTER 4. XML RULE SCHEMA	39
Semantics of the Rule XML Schema.....	39
Rulenodes.....	39
Tag Nodes and Attribute Nodes.....	40
Comparison and Logical Nodes.....	40
Empty Attribute Nodes and Tag Nodes.....	41
Rule XML Outline	41
Rules Implemented Using XML Rule Schema.....	42
Scrolling Should be Limited to One Direction	42
Do Not Use Frames.....	43

The Webpage Should Not Have Pop-ups	44
Do Not Use Markup to Redirect Pages Automatically	45
There Should be a Text Equivalent for Every Non-Text Element	45
Do Not Create Periodically Auto-Refreshing Pages.....	46
Check the Size of Images in the Markup if They Have an Intrinsic Size.....	47
There Should be No Use of Nested Tables.....	47
CHAPTER 5. AUTOMATION TOOL	48
Pre-Processing HTML Page	49
Remove BR Tag.....	49
Remove NBSP	50
Remove Open Tags.....	50
Insert Quote for Attribute Values	51
Rule Evaluator	51
Get Rule	52
Get Rule Description	52
Evaluate Rule.....	52
Process Logical Operator Node	53
Result Comparison with MobileOK Checker.....	54
CHAPTER 6. CONCLUSION AND FUTURE WORK	57
Conclusion	57
Future Work.....	59
REFERENCES	61

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Rules Used in Automation Tool	55
2. Result Comparison of MobileOK Checker and Automation Tool	56

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Evaluation of Rules for Navigation and Links Category.....	15
2. Evaluation of Rules for Page Layout and Content	19
3. Evaluation of Rules for Page Definition Category	25
4. Evaluation of Rules for Image Size Category	30
5. Evaluation of Rules for Tables Category.....	32
6. Evaluation of Rules for Style Sheets Category.....	35
7. Evaluation of Rules for User Input Category	37
8. Rule XML Outline	42
9. Flow Chart for Evaluator Tool.....	49

CHAPTER 1. INTRODUCTION

The global increase in mobile device ownership, evidenced by a 68% increase for smart-phone sales in the United States of America in 2008 according to International Data Corporation [IDC] data, creates great opportunities to distribute and access multimedia content through personalized mobile devices while bringing tremendous challenges. In order to support efficient mobile browsing, some markup languages (e.g., WML and XHTML Mobile Profile) have been developed to specifically support the rendering of information on mobile devices. Furthermore, researchers and designers have summarized previous successful experiences as guidelines that were used to develop mobile-friendly websites. For example, the Mobile Web Best Practices [WMP08], developed by W3C, summarizes the best practices of delivering Web content to mobile users. The Mobile Web Best Practices help design the user interface of websites that can be accessed from a mobile device through the internet. The guidelines also help the designer create webpages that provide a good user experience when browsing the internet from a mobile phone device.

The evaluation of a user interface is one of the vital tasks for the overall user-interface design. The evaluation can take place during any stage of the development process but considering it during the early stages of user-interface development can be cost effective because it can reduce the cost that would otherwise be incurred when completely redesigning an interface.

The design, development and evaluation of mobile interfaces are still an emerging area, compared with traditional interfaces on personal computers. The web interfaces can

be evaluated manually against the guidelines. This task consumes a lot of time, energy and human resources. Therefore, there needs to be some mechanism through which user interfaces can be automatically tested against the guidelines and also provide some feedback to the designers. Feedback can include the interface's shortcomings as well as ways and means by which the interface can be improved. Automatic inspection of mobile interfaces has been a rare research topic. As a useful complement to usability testing and expert review, automation of usability evaluation has various advantages, such as a reduction in cost, less time, an increase in the consistency of the errors uncovered, an increase in coverage of the evaluated features, etc. [Ivo01].

The advantages for automatic evaluation of mobile websites over manual evaluation have been summarized [Lec98].

1. Automation helps in reducing the cost as well as time of evaluation.
2. All the interfaces are evaluated in a consistent manner so that the problems are represented in a similar fashion.
3. There are good chances of increasing the evaluation criteria in the automation tool. In this way, the same interface can be evaluated against multiple features.
4. With automated evaluation, experts in this field are required less frequently as compared to manual evaluation.
5. There might be various alternate user-interface designs for a single application. The automated evaluation can help to compare the different alternate designs and help to continue with the best design.
6. With the advancement in technology and development of tools to perform automated evaluation, the evaluation can be started in the design phase.

In this paper, an extensive and automatic inspection has been conducted on the 46 most popular mobile websites according to the Mobile Web Best Practices using the MobileOK checker [WMC09], a free service provided by W3C that helps to analyze the mobile friendliness of any website. The evaluator applies the MobileOK tests [WMT08] as part of the evaluation process. When all the tests of the MobileOK checker [WMC09] pass, the webpage is said to be MobileOK. A MobileOK checker [WMC09] helps in the creation of webpages that have mobile-friendly content. The evaluation results gathered from the inspection provide valuable information for mobile web designers to improve the usability of mobile browsing. An extensive study of the evaluation results has been performed, and various suggestions have been provided to improve the overall design of the interface. Summarized statistic data can provide alternative solutions as well as identify the common mistakes a web developer can make while creating mobile websites. The results of the analysis can create awareness among the web developers, which can improve the mobile web-browsing experience among the users. Based on the tests performed in this research paper, it was found that the rules under the category “Tables” experience a maximum failure rate. The designers need to be educated about the importance of the table-less design. Table-less design provides flexibility in contrast to the use of tables that provide a rigid structure and layout for web content. Use of `<div>` tags, in contrast to the traditional `<table>` tag, can be utilized to address the problem encountered.

Different mobile devices have different screen sizes, layout structures and styles to represent web content. Use of the MobileOK checker [WMC09] does not provide the capability to evaluate the same guideline in a different manner for different mobile devices.

A tool has been developed to perform automatic analysis of the mobile webpages. The advantage of this tool is that it uses XML, a standard format for representing the guidelines. Another advantage is that the tool provides flexibility to evaluate the same guideline in a different manner for different mobile devices. The focus of this research is creating XML standard rule schema to represent the guidelines that can be automated from W3C. The use of XML gives the extensibility to add more guidelines that represent some additional rules as well as to represent the same guideline in a different way that can be used for different types of mobile devices. The tool accepts the webpage as input along with the guidelines represented in the form of rule schema. The result provides information about which of the guidelines was violated by the input webpage.

The rest of the paper is organized as follows. Chapter 2 discusses related work. Chapter 3 discusses the results and detailed analysis of every rule evaluated for the 46 most popular mobile websites collected from [Can08]. Chapter 4 discusses the XML Rule Schema, followed by Chapter 5 that mentions all the classes that are part of the tool developed and that use the XML rule schema to evaluate mobile websites. Last but not least, Chapter 6 discusses the Conclusion and Future Work that can be done in this area of research.

CHAPTER 2. RELATED WORK

The use and ownership of mobile devices has increased tremendously over the past few years. Mobile devices have become the most sought-after medium of connecting with the world. The ease of accessing information and having the capability to carry out transactions has helped with the integration of mobile devices in the everyday lives of people all around the world. People tend to perform a wide variety of tasks using this portable device. It is turning out to be a revolutionary idea to control various day-to-day activities using a small, handheld mobile device.

This growth has stirred a trend towards the creation of websites for mobile devices. Before the trend, websites were primarily designed for personal computers [Bor07] which had a big size screen to display data and provide options for the webpage. In contrast, for a mobile device, the screen size is much smaller, creating a number of restrictions that have been noticed from time to time. Due to the small size of the mobile screen, horizontal and vertical scrolling is very stressful and takes up a lot of time. Small screen size, in some ways, restricts the amount of information available to the user on a given webpage. Besides these disadvantages, there are other shortcomings pointed out by [Vig08] such as low bandwidth, a low text-input rate and the lack of a pointing device. [Ivo01] suggests that webpages for mobile devices should have less content than the regular version of the websites and that the information should be more contextual to the user. Also, having fewer images and other graphics will help minimize the size of the page, thus reducing the loading time.

A typical webpage has a large number of non-text elements, such as graphics, images, etc. The use of such elements increases the size of the webpage tremendously. The

bandwidth of mobile browsers is limited and tends to affect the user's experience to access a webpage on a small-screen device. The time taken to load a webpage that consists of various HTML elements, Cascading Style Sheets (CSS) style sheets and other non-text elements increases. Also, once a page is loaded, the page needs to be parsed by the browser. If the page is more complex, it will consume a lot of CPU memory as well as performance. In order to improve the user experience of accessing the internet on a small-screen device, the mobile webpage can be run on the server machine instead of the client machine. The server machine can transfer a minimal version of the webpage to client machine [Leh06].

To increase the effectiveness and productivity of using mobile devices to access webpages, there is a need for evaluating the design, development and usability of the webpages. According to [Zha98], usability evaluation is one of the most important tasks that can help improve the design and increase the usability of webpages. [Lec98] pointed out that usability evaluation of a website can be done at any stage of the development life cycle. Evaluations that are performed during the early stages of website development can be more effective because they help predict the usability of the website and can even help a User Interface (UI) designer understand the user's needs and requirements. The later stages of development focus on identifying the various problems of website usage and the steps that should be taken to enhance the user experience, which can be an iterative process, thus further stressing the fact that there is a need to automate the process of usability evaluation. Automated processes/tools can help developers identify whether a website follows the design guidelines or not in a relatively smaller amount of time and without encountering major costs in the process.

A lot of research that centers on the various techniques of usability evaluation has been performed in the last few years. [Nie90], in their work, provide a lot of emphasis on *heuristic evaluation*, which is a technique that requires UI experts to perform a thorough analysis of the webpage and to identify the various difficulties that a user might face while utilizing that website. Such tests can help gather a lot of information about whether a webpage satisfies user requirements and can identify areas of improvement. This kind of evaluation is promising, but has various limitations. One of the biggest hurdles for performing heuristic evaluation is that there are very few people who have the necessary UI experience to carry out such evaluations. The UI improvement techniques are difficult to apply before an interface exists; the techniques are also difficult to apply in the later stages when the UI is completely ready because it is complicated to accommodate any changes.

[Bro88], [Smi86] and [Shn86], in their work, highlight *published guidelines*, yet another method used to evaluate the user interface. There is a set of rules used to check against the webpage and to certify whether a webpage satisfies the rules.

[Lew90a] and [Lew90b], in their research, emphasize the *cognitive walkthrough* method, a combination of software walkthroughs and learning different aspects of the UI by exploration. In this technique, the UI developers walk through the entire general task that a user will perform on the website. The results gathered are compared with the users' knowledge and the goals they are trying to accomplish. Also, the inconsistency between the users' expectations and the gathered results is examined.

Alternative means of evaluating interfaces have been proposed, which try to structure the evaluation process so that interface developers, not UI specialists, can carry

out the evaluation, potentially increasing the number of people who can do evaluations and avoiding some of the problems mentioned earlier.

[Jef91] did a good study comparing the various methods and established that usability testing based upon heuristic evaluation is very expensive and time consuming, although the evaluation tends to identify most of the very serious user-interface problems. The high cost is due to the fact that it is difficult to find people with expertise in this domain. The performance of usability testing based on published guidelines is an improvement on the heuristic method because there is a definitive set of rules that can be compared against the webpages. Most of the common mistakes that a developer makes while creating the websites, can be identified using this approach. The cognitive walkthrough method can also be used by software developers; the method can help define users' goals and assumptions, but can miss the most common and general recurring problem.

[Nie93] and [Dix98] have presented an extensive survey of usability evaluation methods, organized according to a new taxonomy that emphasizes the role of automation. They talk about some aspects of UI evaluation, such as *Capture*, *Analysis* and *Critique*. Capture includes capturing or collecting relevant data, such as how much time it takes for a task to finish. Analysis includes analyzing a given user interface with respect to some factors and reporting the problems encountered in the interface. Critique represents providing some suggestions to improve the interface.

[Ivo00] proposed a taxonomy that explores various aspects of evaluation. It is represented as follows: Non-automatic, Automatic Capture, Automatic Analysis and Automatic Critic. The Non-automatic evaluation, as the name suggests, is to rely

completely on manual evaluation that makes use of experts. The others consist of methods and techniques that can be employed to satisfy different approaches of usability testing by employing automatic methods of identifying problems and performing analysis.

As pointed out by [Arr06], performing evaluation manually is a very difficult task due to the large amount of information present on different websites. Automatic evaluation is always considered a better approach. The tools that perform automatic evaluation can be categorized based on the type of reports they provide. One of the types is the kind of tools that inform what rules the website is not following. The other could be that, in addition to providing faults, the tool also helps the designer improve the design of the mobile websites' UI. Another type of classification is based on whether the tool runs on the web server or can be executed from a local computer. Also, the tools might use a set of static rules or have the capability to include the set of rules that can be added when required or when new guidelines are defined by W3C. In order to perform automatic evaluation, there needs to be a set of rules that can help evaluate the user experience of accessing the internet and websites through a mobile device.

The guidelines can be used to automatically verify web accessibility in a way to guarantee the web-accessible design, one of the biggest challenges that have been faced by many designers across the globe. The web designer must have immense knowledge, experience and awareness of various guidelines. Because of the changing web technology, the set of guidelines needs to be updated on a regular basis; new sets of guidelines keep on being added, so the tools needed to verify accessibility of a web design should incorporate all these changes.

There are large numbers of tools that are already available to verify the accessibility of any website. The tools need to observe, analyze and interpret the website characteristics themselves. These tools can be of three different categories: Automatic Capture, Analysis and Critique. Automatic Capture tools, as the name suggests, collect different user and system information. Analysis tools are broadly characterized into two kinds. One of them is called general evaluation tools; these tools help to mark different accessibility problems for any website by producing a report and checking the HTML code against the guidelines specified by WAI/W3C. The other kind is known as focused evaluation tools which focus on testing only some of the accessibility aspects.

There was much debate about what constitutes good website design [Nie00, Rat96]. Guidelines for user interfaces were high level, and designers had difficulties following and implementing them as stated by [Bor96], [Des90], [Low92], and [Smi86]. Thus, there was a need for a set of rules that can be used for the automatic evaluation of webpages used to access the internet with mobile devices. This led to the release of Mobile Best Practices 1.0 in 2006. These practices, in turn, evolved into Web Content Accessibility Guidelines [WCAG99]. These rules started to cause mobile webpages to be constructed in such a way that these pages are simpler, have less content than the regular version of websites, and have fewer images and other graphics in order to decrease the loading time. Using this set of rules, much research was done to model the rules and build tools to automate the process of usability evaluation. These best practices determined a set of basic MobileOK tests that can help improve the design and interface of such websites.

In their research, [Sch98] focused on developing software tools and techniques for evaluating the usability of websites. Their approach includes two types of tools and

techniques. First, usability awareness tools are used by website designers and developers who may not be familiar with usability issues. Second, Web Usability tools are developed for use by usability professionals. These tools are concerned with increasing effectiveness by focusing on speed evaluation, a wider audience for usability testing and built-in analysis features.

The research that has been done for this paper focuses on the analysis of the 46 most-common websites using MobileOK checker and provides useful information to web designers regarding the common mistakes made during the design and creation of mobile websites. Also, an evaluation tool has been created; it uses an XML structure to represent the guidelines. XML is an open standard that can easily be modified, which helps when updating the guidelines as well as adding a new set of evaluation criteria, thus providing flexibility and extensibility. The tool accepts the rule schema, as well as the mobile webpage, and provides a report stating which rules passed and failed. This information can be used for analyzing and improving the design of webpages.

CHAPTER 3. EVALUATION BASED ON MOBILE OK CHECKER

The MobileOK checker has been used to evaluate a list of most the popular mobile websites [Can08]. We wanted to evaluate the current market trends and see how the results of this research help designers/developers know the current pattern of website development. The results created a deeper understanding of web development for mobile devices while pointing out a few devices which have higher violation rates and need to be addressed in further development. The MobileOK checker uses a set of 24 rules from the W3C Mobile Web Guidelines. The results provided by the MobileOK checker have been used to understand the significance of each rule and to help the designers provide valuable information to improve the design and interface for mobile websites.

A test resulting as “*Pass*” specifies that the rule has been successfully followed. A test resulting as “*Fail*” indicates that the rule has been violated by the website. A test resulting as “*Warning*” illustrates that the tool was not able to determine 100% that the content for the webpage has failed for a particular rule or may also suggest that the web content is very close to violating a guideline.

The rules have been divided into multiple categories. These categories are defined in the W3C Mobile Web Practices [WMP08] document. The categories are as follows: Navigation and Links, Page Layout and Content, Page Definition, Image Size, Tables, Style Sheets and User Input. A detailed evaluation is given in the rest of the chapter.

Navigation and Links

The Navigation and Links category consists of the following rules: Auto Refresh, External Resource, Pop-ups, Image Map and Redirection.

Auto Refresh

“Do not create periodically auto-refreshing pages, unless you have informed the user and provided a means of stopping it.” [WMP08]

Auto refresh means the ability of the page to automatically reload itself in some intervals of time. It causes a lot of accessibility problems. If the auto refresh is set to true for a webpage, it causes an extra cost for the page to reload. Also, more memory is utilized. In order to avoid these extra costs, the mobile webpages should not have Auto Refresh activated.

Similarly, the pages might have the redirection set to true, which may further slow the loading of webpages on a mobile device thus the webpages should not have any markups which can redirect pages automatically.

External Resources

“Keep the number of externally linked resources to a minimum.” [WMP08]

An external resource means any objects/images that are referenced in the mobile webpages. The rule focuses on determining that use of external resources should be minimized as much as possible because accessing each external resource is considered a separate request.

In order to evaluate this rule, the total number of externally linked resources is calculated. If the number exceeds 10, the tool gives a warning, and if the number exceeds 20, the rule tends to fail for any given website.

Pop-ups

“Do not cause pop-ups or other windows to appear and do not change the current window without informing the user.” [WMP08]

A pop-up is a graphical user interface display area. It is usually a small window that suddenly prompts the user in the foreground of the visual interface.

In order to test this functionality, the MobileOK checker finds the “target” attribute on the links, and if the “target” attribute is present, the MobileOK checker will see if the attribute has a value different from “_self,” “_parent” or “_top.” If the value differs, then the test is marked as failed.

Redirection

“Do not use markup to redirect pages automatically. Instead, configure the server to perform redirects by means of HTTP 3xx codes” [WMP08]

Automatic redirection significantly increases the response time to load a page. Each redirection causes a request to the server and a response back from the server. The increase in response time leads to poor performance and increased cost.

In order to evaluate this rule, the tool checks for the presence of a *meta* tag with a content attribute having a value for a different URL in the HTML page header. Such a tag is defined in the following way:

```
<meta http-equiv="refresh" content="a different url">
```

Image Maps

“Do not use image maps unless you know the device supports them effectively” [WMP08]

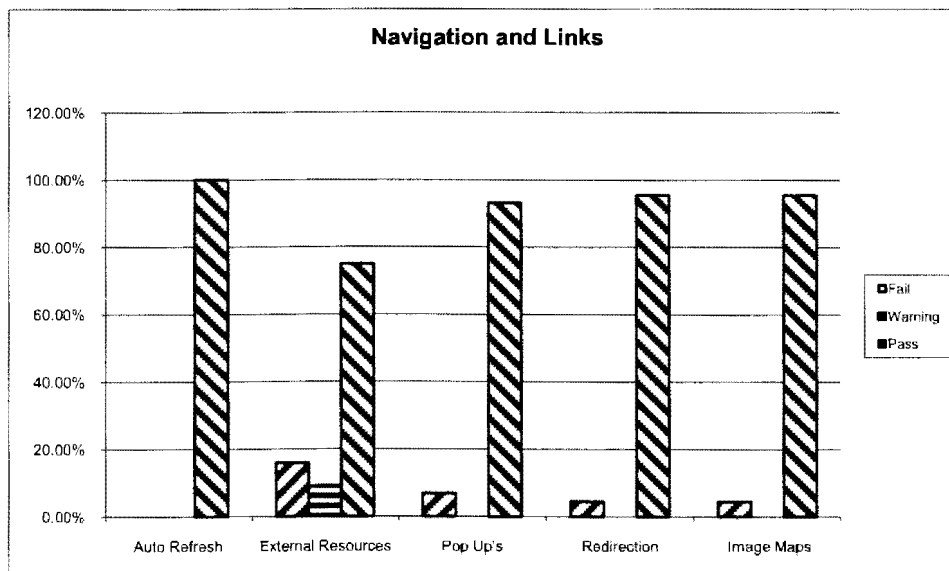
An image map is a list of coordinates corresponding to a particular image, whereby areas described by these coordinates hyperlink to different URLs. These area maps can even be programmed to run a script (e.g., JavaScript). For example, in a map of an office building, each room or section can link to a page where a more detailed map of the room or section is provided. All the devices do not support this functionality.

In order to evaluate this rule, we check if an input element with the type attribute set to "image" is present. If and "image" attribute is present, the rule fails. We also check each tag. If a "usemap" or "ismap" attribute is present, the rule fails.

Discussion

Under this category, all the rules have a high pass percentage rate. The below Figure 1 charts out the results and is followed by detailed discussion.

Figure 1. Evaluation of Rules for Navigation and Links Category



The rule that has been violated the most is the use of External Resources; this rule has a failure rate of 15% because the number of externally linked resources exceeds 20.

However, about 9% of the remaining websites give a warning for this rule because the number of externally linked resources exceeds 10. With more external resources associated with a webpage, there will be a greater time and cost to load a webpage, leading to an increase in response time. In order to minimize the response time, we need to minimize the external resources. Designers need to remember that the use of external, embedded links should be minimized as much as possible. Instead of using external links, the websites can include a lot of informational content that is self explanatory such that there is no need to embed additional external links, which will help the users to gather sufficient knowledge at one place rather than having access different links at the same time.

The use of pop-ups on mobile webpages has a substantial failure rate of 6.82%. Pop-ups literally mean something popping from the interface, and separate windows for inputting data in some mobile devices do not fall under pop-ups. Pop-ups, in this context, have many downfalls and will eventually end up confusing the user, adding costs and delaying the interaction. Most mobile devices only support one window per session, and attempting to open more than one window in a single session will display unpredictable results. It is almost impossible for a pop-up scenario on a mobile device because most mobile devices do not support this functionality. Although a number of designers are aware of this fact and, therefore, have designed their websites in such a way, there are a few who need to be educated about the importance of this rule and the impact it can have on anyone going through that experience. An alternate solution for using pop-up windows is to include inline messages on the page itself.

The redirection rule has a failure rate of 6% that cannot be ignored. Automatic redirection increases the response time and provides a bad user experience. Even worse,

one redirect leads to another, causing a loop of redirects in some cases. Furthermore, the importance of redirection can be more understood in relation to transaction-based websites where a user might lose some important content if a redirect happens in the middle of a transaction. Most designers have a good understanding of this problem, and the rule has been followed to a large extent. Instead of automatic redirection, server-side scripting should be used to perform redirects to another page.

The advantage of using this approach is that the redirects will happen on the server side without any performance issues on the client side because there is no need to download any page or script that handles the redirect. The method is also supported by all browsers.

Most mobile devices allow user keys for left, right, up and down navigation, which is usually enough for a user to explore maps, defined as geometric shapes, on an image. To do an effective, fast navigation, the device must be a pointing device. Many mobile devices do not support a pointing device. Image maps cannot be supported by such devices. The image-map rule has been followed by most mobile websites with a high passing percentage of 96%.

To replace image maps, only small images can be displayed by breaking larger images into smaller sections and working with them separately. A list of links with descriptive text can also be used.

It is noted that the auto refresh has been followed by all the websites, with a pass percentage of 100%. Thus we can ignore auto refresh rule in evaluation process.

Page Layout and Content

The rules that the MobileOK checker evaluates under this category are as follows:
page limit size and graphics for spacing.

Page Size Limit

“Ensure that the overall size of page is appropriate to the memory limitations of the device” [WMP08]

The rule ensures that the page size should not exceed 20 KB. This size includes all the external resources, CSS files, images, script, etc. It becomes difficult to load a page that is greater than 20 KB on a mobile device due to network and connection limitations.

Graphics for Spacing

“Do not use graphics for spacing” [WMP08]

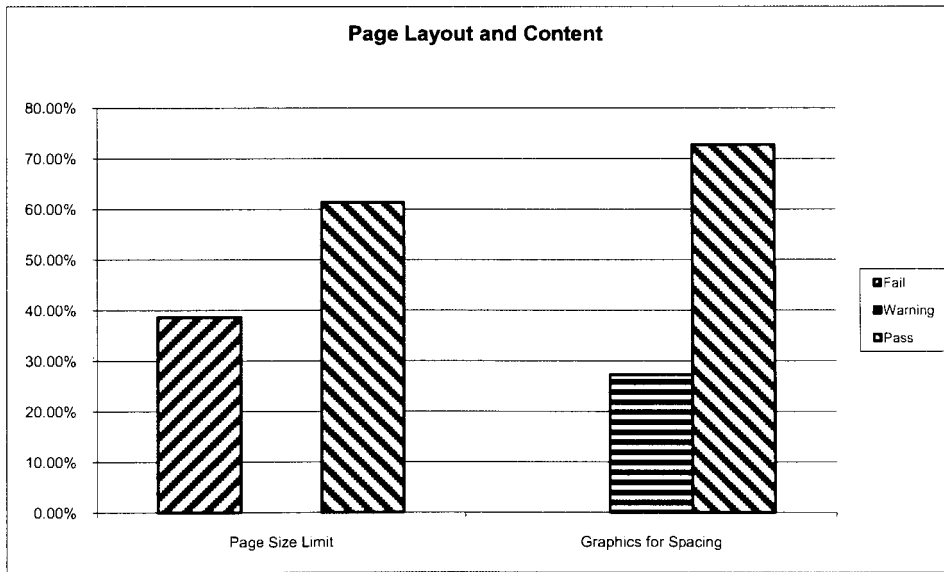
This rule implies that mobile websites should not use graphics in order to position and fix the layout because not all mobile devices are able to support this functionality. In the case of mobile devices that do not support this functionality, the webpage will not be displayed correctly.

The test is configured in such a way that it will give a warning to the user if small-sized transparent images (2*2) are detected. In the case of large images, the tool will result in a failure.

Discussion

Under this category, all the rules have a high fail percentage rate. The below Figure 2 charts out the results and is followed by detailed discussion.

Figure 2. Evaluation of Rules for Page Layout and Content



High percentage (61%) of websites have an overall page size less than 20 KB, and we can successfully conclude that these websites have passed the rule. The designers need to remember the size of the page while designing. It becomes very inconvenient to load pages with a greater size because mobile speeds are not high while accessing the internet. On the other hand, the page size should not be too small because that will require sending multiple requests from the server. Multiple requests add to the delay for loading a page. However, users might face difficulty while accessing the remaining 40% of the websites with a page size higher than 20 KB.

Designers need to minimize the use of external resources, scripts, CSS as well as images or to use them in such a way that the overall size does not exceed the 20-KB limit that is defined in the default delivery context. Developing webpages that are text based will be an alternate solution and help in reducing the page size.

For websites that have been evaluated, we have a 0% failure for the graphics for the spacing rule which means that none of the websites used spacing images which have a size of more than 2 pixels for the width and height. About 28% of the websites give a warning for this rule because small, transparent images with a width and height less than 2 pixels have been detected. The remaining 72% of the websites have passed this rule. The designers need to minimize the use of images for spacing purposes. Reducing the images will further lead to a reduction in page size and will help developers keep the page size limit below 20 KB.

Page Definition

The Page Definition Category is comprised of the Page Title, Minimize, Content Format and Support, Character Encoding Use and Support, Caching, Valid Mark Up, Non-Text Alternatives, and Objects or Script.

Page Title

“Provide a short but descriptive page title” [WMP08]

The rule states that every webpage should have a page title associated with it. If the page title is missing or the value of title attribute is empty, the rule fails. On the other hand, the title should not be too long that it contributes to the size of the page. The page title helps identify the webpage.

In order to test this rule, the tool checks for the “title” attribute on a webpage. If the title attribute value is an empty string or the page title attribute is missing, the tool gives a failure. In case the attribute is present with a title value which is not an empty string, the tool passes the rule.

Minimize

“Use terse, efficient markup”[WMP08]

This rule ensures that the webpages designed for mobile devices should minimize the presence of white-space characters. There are some networks that remove the white-space characters from the webpage while loading the page from the server, but not all network connections do the same.

In order to test this rule, the tool counts the number of white-space characters and also adds to it the number of characters of XML comments. If the number of extra characters is greater than 10% of the total characters, the tool gives a warning, and if it exceeds 25% of the total number of characters, the tool gives a failure for that webpage.

Content Format and Support

“Ensure that content is encoded using a character encoding that is known to be supported by the device” [WMP08]

This rule explains that the mobile websites should send the content in such a way that it is compatible with the mobile device. In order to find the supported formats, the websites should use the device-profile information in the form of the HTTP Accept header, HTTP User Agent header, etc. In order to evaluate this rule, we need to check the Multipurpose Internet Mail Extensions (MIME) type of content using various devices. All the embedded images and CSS in a webpage should be delivered in a way that is considered as part of the default delivery context. The CSS should be syntactically correct, the images and other media embedded in the webpage should be present in a format that is known to be acceptable by the device or considered as part of default delivery context.

Character Encoding Use and Support

“Ensure that content is encoded using a character encoding that is known to be supported by the device. Indicate in the response the character encoding being used.” [WMP08]

All websites need to specify the character encoding that is being used. The encoding can be specified in many ways, such as using the HTTP Content Type header in the HTTP Response. For example,

```
Content-Type: text/html; charset=utf-8
```

Also, the XML documents can have an encoding declaration in the form of

```
<? xml version = “1.0” encoding= “UTF-8”?>
```

The W3C Guidelines [WMP08] recommend the use of standard UTF-8 encoding. The bandwidth that is required to transmit content depends on the character encoding used.

Caching

“Provide caching information in HTTP responses” [WMP08]

Caching is defined as the temporary storage of the response, such as CSS files, images, HTML pages, etc., that is received when a webpage is being accessed. When a request for the same URL is received, instead of getting the response from the server, the local copy that is saved can be used to render the webpage. Caching is important because it reduces response time and reduces the cost to load a page, thereby increasing the performance. This way, it also reduces the web traffic because the number of requests being sent to the server decreases tremendously. The rule can be detected by inspecting the presence of cache headers in the response that is received.

The presence of *cache-control: public* in the response header means that the caching of information is allowed by different devices. Similarly, *cache-control: private*

can be used to allow sharing by the same mobile device, and *cache-control: no-cache* is used to disallow caching. There can be also cases when there is no caching information present in the response header.

Valid Mark Up

“Create documents that validate to published formal grammars” [WMP08]

Document Type Definition (DTD) is defined as the grammar of an XML page. Once a grammar is defined, various HTML pages are written on the basis of the grammar already declared. This technique is used in determining whether an HTML page is a valid page. In order to find the validity, the grammar that the page is built from should be specified at the start of the HTML page to be used by the checker.

This rule explains that the HTML mobile webpages should conform to HTML, XHTML, MathML or SVG format. Failure to do so can result in an improper presentation, or rendering, of the webpage on a mobile device. The valid mark up rule also includes the exclusion of the deprecated features of W3C technologies. Such a use of style sheets is recommended instead of using the font attribute as part of the webpage.

The tool checks for the presence of a *doctype* element in the header of the HTML page that follows the published grammars. The *doctype* is used at the start of the webpage and specifies which version of HTML should be used by the validator to verify the syntax of the page. An example of the same would be as follows:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

The above *doctype* declaration specifies that the HTML page follows the strict XHTML format.

Non-Text Alternative

“Provide a text equivalent for every non-text element” [WMP08]

The rule means that any non-textual element on a mobile webpage needs to have text associated with it. In order to test this rule, we need to make sure that the “alt” attribute is present on the image tag. With the alt attribute, even when the images do not load properly, the associated text will be displayed on the screen, making the content readable and understandable for the user. Also, the alt attribute should not be empty. In order for the rule to pass against the webpage, the alt attribute should be present with a value associated with it.

Objects or Script

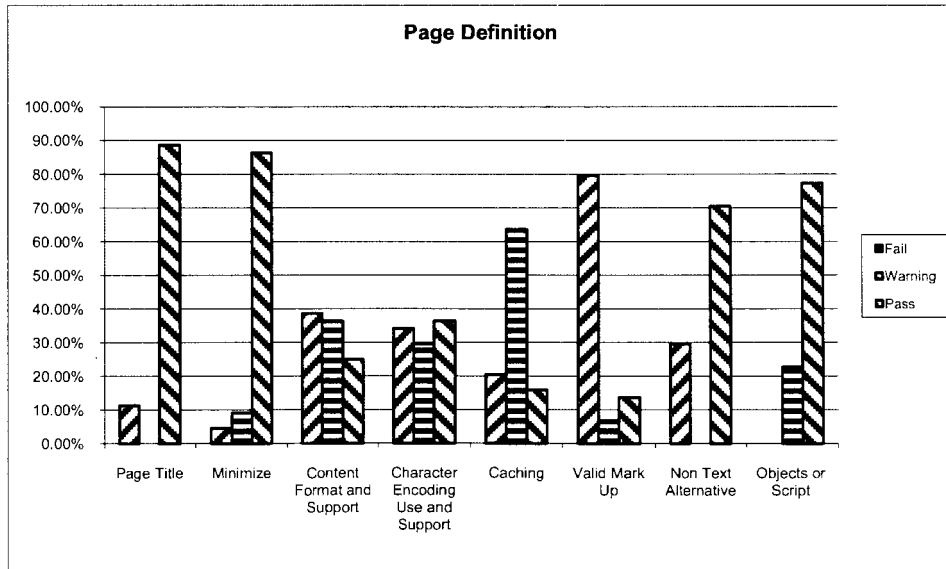
“Do not rely on embedded objects or script.” [WMP08]

This rule inhibits the use of objects or script elements. In order to evaluate this rule, the tool checks for the presence of applets. If applets are detected, the website fails for this rule. Similarly, if the script element is present, a warning is returned. For each <a> and link element present on the webpage, the tool checks if the “href” attribute starts with a Java script element. If it does, then the tool returns a failure for this rule. If any element has an event attribute associated with it, the tool returns a warning.

Discussion

Under this category, the rule that has been violated the most is the Valid Mark-up Rule. It has a failure rate of nearly 80%. The below Figure 3 charts out the results and is followed by detailed discussion.

Figure 3. Evaluation of Rules for Page Definition Category



The designers need to give a tremendous amount of attention in specifying a valid doctype for the HTML pages. The HTML mobile webpages should conform to the HTML, XHTML, MathML or SVG format. Failure to do so can result in an improper presentation, or rendering, of the webpage on a mobile device. The valid mark up rule also includes an exclusion of the deprecated features of W3C technologies. Such use of style sheets is recommended instead of using the font attribute as part of the webpage.

This content format support rule has a substantially high failure percentage of 38% and a pass percentage of only 25%. There is also a significantly high rate of websites for which the tool gives a warning. Overall, the rule is not really being followed to a large extent by the designers who create and build the websites. The rule needs more awareness about how this aspect can be taken care of while designing the websites.

In regular terminology, character encoding means how the characters will be represented as bits when stored in a computer file. In the evaluation of 46 mobile websites, 34% of the websites failed the character-encoding rule because the HTTP Content Type

header does not specify any character encoding. The XML documents for these websites neither provided UTF-8 nor did they have any other encoding in place. There were 29.55% of the websites that gave a warning because they specified that the encoding is being used, but there were no traces that the standard UTF-8 is being used. UTF-8 is supported by all mobile devices, so the website will load correctly without error. There might be instances that the application fails to load properly if the standard UTF-8 is not supported by the website. The prime reason behind this failure is that UTF-8 allows all characters in the document character set to be included. Thus, as a solution to get rid of such errors, it is highly recommended for developers to make use of UTF-8. Making use of UTF-8 encoding will also help in creating consistency among websites that are designed by different developers.

This Non-Text Alternative rule has been passed by 70% of the websites and 30% of the websites have failed because the alt attribute is missing from the image tags in the HTML webpage. The non-text alternative rule is one of the most important rules that should not be ignored in any circumstance. There might be mobile phone devices that have graphics turned off or Java script disabled. In such cases, the presence of alt attribute helps understand the context of the image missing from the webpage. It takes a large amount of time to download images on a mobile device. The total time for displaying a single image includes the time it takes to download and the cost of displaying the image on a mobile device. In the case when the images do not load correctly, if we have non-text elements associated with images, the webpage can still be read by the user. Thus, we highly encourage the use of alt tags and even recommend them to be used as an industry-wide best practice.

There is a significantly high percentage of websites where the response header specifies a no cache attribute, thus none of the elements, such as CSS, HTML pages, webpages, etc., are in the memory. The entire website needs to be reloaded for each request. The tool gives a warning for 64% of the total websites analyzed; 20% of the websites fail because there is no caching information present. Only 15% of the websites completely pass the rule of caching. The results lead to the conclusion that there is definitely not much awareness about this rule among web developers. A large amount of network bandwidth can be saved if significant importance is given to this rule. When a page is requested, most time is spent in loading all the related elements of the webpage, such as external style sheets, scripts, images, etc. Caching allows fewer HTTP requests that are sent to the server in order to render the page, thus reducing the response time. The cached element allows the request to load just the content of the webpage a second time, reusing the cached components to render the remaining pages of website.

Even though the page-title rule is quite straightforward with obvious implications, we still observe that 11% of the webpages fail this rule. We highly suggest that developers spend some time adding relevant and precise page titles which provides users information about what they are accessing, thus enhancing the user experience. Little effort and time can yield a multitude of results.

Out of a total of 46 websites that have been tested for the rule to minimize, even though only 5% of them fail, we observe that another 10% have a warning associated with them. Developers should spend time clearing out the extraneous characters from the webpage. The objective is to minimize the XML size, which decreases the page's load time. All the unnecessary white space should be removed from the page. Representation of

a style for different elements should be specified in a separate CSS file rather than inline using the style-class attribute. Using the style sheet decreases the page weight and size as well as reduces the load time.

The rule of having objects or script failed for none of the websites and has a high passing rate of 78%. A high percentage (28%) of the websites returns a warning. The warning is returned because the tool finds the script attribute, and some elements of the webpage are detected to have an event attribute. There are high chances that the scripts are unable to load correctly on small-screen devices, and it is even more difficult to load external plug ins in order to support the external scripts. In addition, the use of scripts also increases the power consumption and decreases the battery life of mobile devices. The page should be designed in such a way that it is also usable in the text-only mode so that, in cases when the mobile device does not support objects or script, the page is rendered correctly and the user is able to make use of all the features of a webpage. The designers need to make sure that, if objects or script are being used, there is a text equivalent, such as the use of a NOSCRIPT element, so that the page can still be usable if the scripts do not load.

Image Size

The Image Size Category has two rules associated with the size of the image: Image Resizing and Image Specify Size.

Image Resizing

“Resize images at the server, if they have an intrinsic size” [WMP08]

This rule states that the images used for a webpage should be resized at the server. Resizing the images has the advantage that it reduces the amount of data transferred across the network and also decreases the processing time consumed to rescale the image to fit a particular mobile device.

In order to evaluate this rule, the tool checks the width and height attribute of the image node. If the width or height attribute exceeds the actual dimensions of the image, the tool will give a warning, and if the height or width attribute of the image is less than the actual dimensions of the image, the tool results a failure.

Images Specify Size

“Specify the size of images in markup, if they have an intrinsic size” [WMP08]

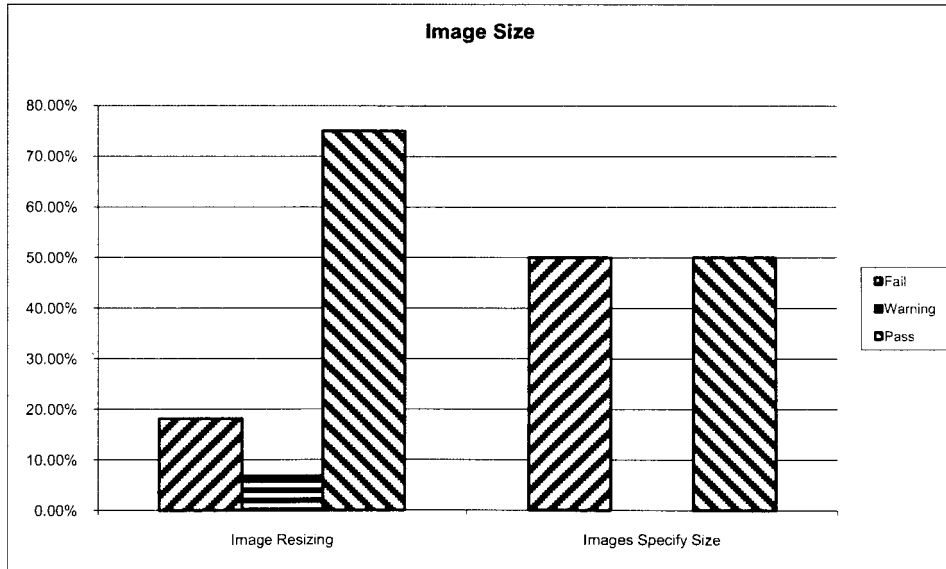
This rule states that designers need to specify the image size because most of the images that would be used for mobile webpages have a very small size because they are developed to be viewed using small-screen devices. It further helps the mobile browser to load the pages more efficiently if the size of the image is well known in advance.

In order to test this rule, the tool checks the presence of width and height attributes for all images. If any attribute is missing, the tool returns a failure. The size of the images needs to be specified in pixel size. If the size is not specified in pixels, the rule fails for that particular website.

Discussion

Under this category, all the rules have a high fail percentage rate. The below Figure 4 charts out the results and is followed by detailed discussion about rules for Image Size Category.

Figure 4. Evaluation of Rules for Image Size Category



In the evaluation conducted using the MobileOK checker on 46 websites, 50% of the websites failed the rule for specifying the size of the images because either the height or width attribute was missing for the image elements. The exact image measures help the browser to load images correctly and also avoid reloading the page when it has been loaded once. Specifying the height and width attribute is not much additional work on the developer's side, but the specification increases effectiveness. This task should be part of best practices followed by a developer and on an industry-wide level. Providing size information for the browsers ensures fast page rendering by all browsers.

Even the image resizing rule has a failed with almost 25% of websites either having a failure or warning message. Thus, it seems that a number of mobile website designers tend to ignore this rule, leading to the failure for 18% of the websites with the reason that the height or width that is defined for an image is less than the actual dimension of the image. The entire image will still be downloaded, and a lot of network bandwidth will be

wasted fitting the image to the smaller size specified on the webpage. In order to avoid this problem, the actual dimensions of the image should be used. The remaining 7% of the websites return a warning for reasons where the height or width attribute is more than the dimensions of the image. These incorrect or missing attribute specifications can cause slower loading of the page images, thus adversely affecting user experience. Specifying the attributes properly should be adopted as an industry-wide best practice. The rule also sheds light on the fact these common mistakes that a website developer might make while creating the websites that can be reduced if awareness is created among the community.

Tables

The tables category contains all the rules associated with the use of tables on a mobile webpage. The rules are Table Alternatives, Table Layout and Table Nested.

Table Alternatives, Table Layout and Table Nested

Tables Layout: "Do not use tables for layout" [WMP08]

Tables Alternatives: "Where possible, use an alternative to tabular presentation" [WMP08]

Tables Nested: "Do not use nested tables" [WMP08]

Each rule ensures that the mobile webpages need to be designed without the use of HTML tables. The use of tables in designing webpages can sometimes result in having the user scroll the page horizontally. The approach of using a nested table, i.e., a table element present within a table, is not recommended.

In order to test these rules, the tool searches for the presence of table elements within a webpage. If it finds one, a warning is thrown. However, if the webpage has a table

element within another table element, it results a failure because there is a use of nested tables for the webpage.

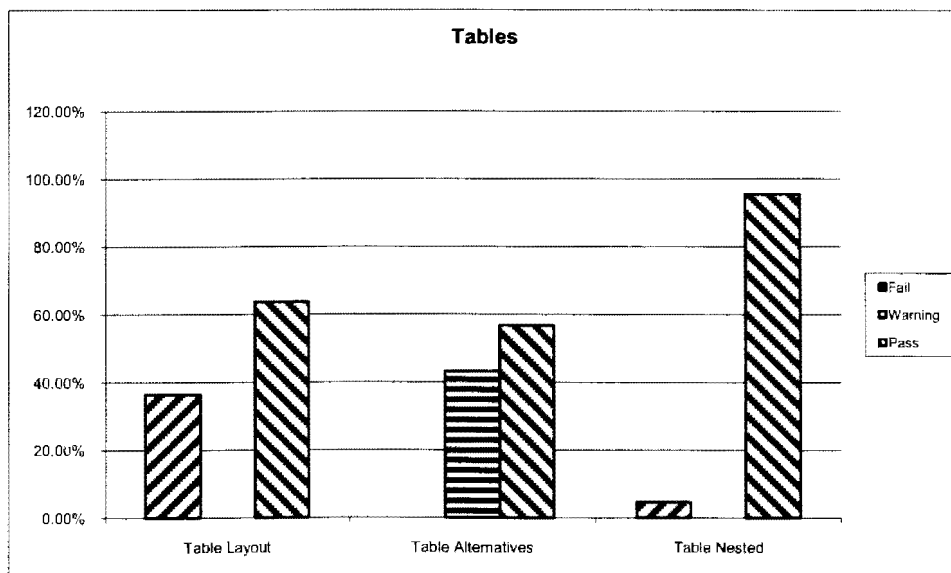
If a table element is present with, at most, a single row, an error is indicated.

Similarly, if no row of the table contains more than one column, then the tool returns a failure. Last but not least, for any given nested `<td>` element, the tool returns a failure, if it contains only image of size less than 2*2.

Discussion

Under this category, the rules have a moderate fail percentage rate. The below Figure 5 charts out the results and is followed by detailed discussion.

Figure 5. Evaluation of Rules for Tables Category



The tables are used by most designers to create a layout and define the overall design of a webpage. This practice is common and poses little harm if the website is being designed to be accessible from desktop computers. However, websites designed for mobile devices are restricted by the small screen sizes of the mobile devices. Table-less design are becoming popular these days because one does not need to specify the height and width of

the page. All the page content can be controlled using `<div>` tags that adjust well to the kind of screen used for webpage display. For example, if the webpage is loaded using a mobile device with a small screen, the page will adjust to the limited space and render correctly. Similarly, if the webpage is requested from a regular desktop computer, the page will try to adjust to the available space accordingly. Moreover, making use of tables fixes the width of the page so that, when the mobile webpage is loaded on a mobile device, a scroll bar appears to traverse the width of the page. The designers should be aware of the use of a table-less design. Almost 35% of websites have failed the Tables layout rule. The designers should include in their testing if there is any table row that does not contain more than one table column element or if an image of very small size is used in any table column element as well as testing if the table contains more than one table row of elements.

The experts in this field focus on the table-less design, getting rid of the unnecessary `<table>`, `<td>` and `<tr>` tags that increase the page size, and design the page layout using cascading style sheets. The increasing trend of table-less design also gains importance because the page-loading decreases tremendously. The table less format helps designers to create more complicated webpages in contrast to using tables that make the website very complex, rigid and inflexible in its format. The table less format also helps increase the search ability for the search engines due to smaller page size. The overall cost to change the look and feel of the website is low because only the CSS can be modified and can incorporate the new branding elements within the website. Similarly, the cost of redesigning a website is also less for a table-less design in comparison to a table-based design.

A lot of designers should be made aware not to make use of nested tables. Even though few designers use nested tables for mobile webpages, 5% of the websites have still failed this rule.

Style Sheets

Style Sheet Support and Use

“Use style sheets to control layout and presentation, unless the device is known not to support them” [WMP08]

“Organize documents so that if necessary they may be read without style sheets” [WMP08]

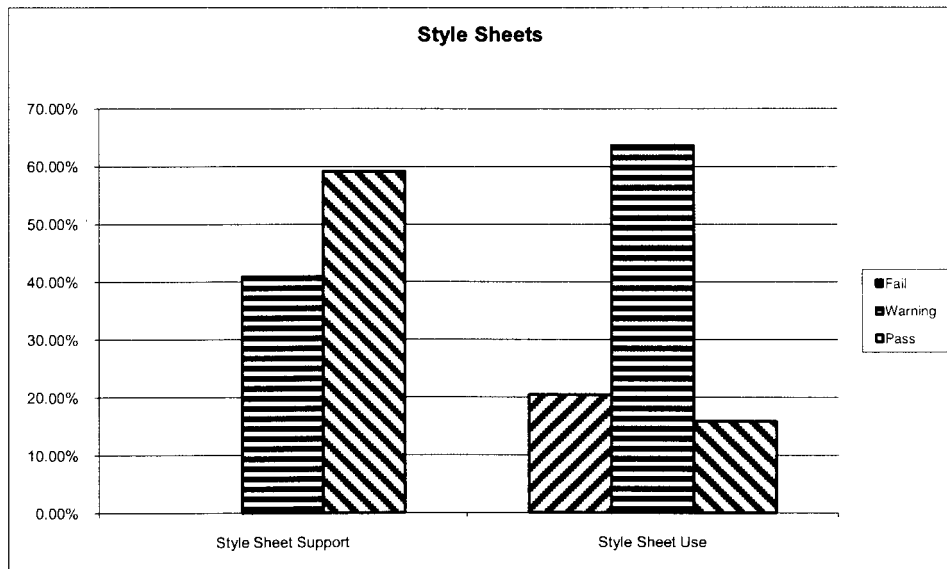
A style sheet is an external file that is referenced by the webpage and lists all the style information for the entire webpage, such as the fonts to be used, colors, background images and margins to be applied. A style sheet is also used to give a consistent look and feel to all the pages of the websites, further avoiding the specification of redundant information on all webpages.

In addition, the CSS media rule [WMT09] can be used to specify different style behaviors using different devices. The media rule is used to differentiate if the page is being accessed from a mobile device or a computer, with values as “handheld” and “all” respectively. In cases when handheld is not specified, the mobile browser can download some other external style sheets that may not apply for small-screen devices. In order to test these rules, the tool finds the presence of style elements on the webpage and gives a failure when style elements are found on the webpage.

Discussion

Under this category, the rules have a moderate fail but high warning percentage rate. The below Figure 6 charts out the results and is followed by detailed discussion.

Figure 6. Evaluation of Rules for Style Sheets Category



This rule needs attention from designers because 20% of the websites report a failure for the use of style sheets and because a large 65% further report a warning. The rules specify that the webpages need to be designed in such a way that they are able to load correctly and in a readable way even when the mobile device does not support external style sheets. The style element can also be part of the webpage; that format increases the page size, so we need to avoid that choice. Alternatively, an external style sheet should be used. Making use of a style sheet has another advantage; when the page is requested the first time, some mobile phones have the capability to cache the externally linked style sheet. In that case, all subsequent requests will not put a large amount of burden on the network because the external style sheets will not be loaded again. However, the amount of burden caused by loading of external style sheets is dependent on the mobile device being

used. In some cases, the mobile devices do not have the capability to cache and/or load style sheets.

Therefore, it becomes very important that the page should be readable even when the style sheets are unable to be read by mobile devices. We need to keep in mind that the style sheet should be as small as possible. We need to make sure that all information in the external style sheet is optimized to the highest level. Only styles that are being used should be included in the style sheet. Any extra information needs to be discarded. Only relevant elements used on the webpage should be part of the style sheet, thus reducing its size.

User Input

Provide Defaults

“Provide pre-selected default values where possible” [WMP08]

The rule specifies that the designer should provide default values for all user controls that are utilized on a webpage. The tool checks for each radio-button group present on a webpage; exactly one input element in the group should have its checked attribute set to true. If this is not the case, then the rule gives a warning for the webpage. A similar check is also applied on each select element where, if we have more than one option with the selected set to true and the multiple attribute is not set to multiple, the rule gives a warning. The prime idea is to make sure that all empty controls, textboxes, radio buttons, checkboxes and other input controls are handled in an efficient way.

Default Input Mode

“Specify a default text entry mode, language and/or input format, if the device is known to support it” [WMP08]

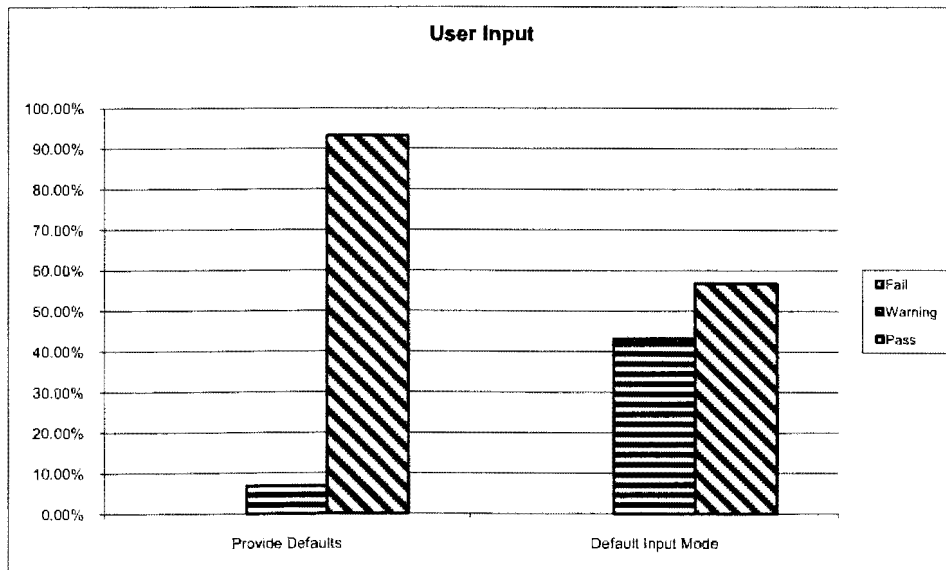
Due to the small screen size of mobile devices, the mobile websites should be designed in such a way that there is minimum use of data entry by user. The user interface of the mobile websites should maximize the use of radio buttons and selection lists in contrast to text boxes and text-area input controls.

In order to test the rule, we need to send a request and check the response. If the response contains text and text-area input elements, input mode attribute should be considered.

Discussion

Under this category, the rules have a high warning percentage rate. The below Figure 7 charts out the results and is followed by detailed discussion.

Figure 7. Evaluation of Rules for User Input Category



It is shown that a significant percentage (43.18%) of the websites gave a warning for the rule of providing default input mode, and the primary reason behind this finding is the missing input mode attribute for the text and text-area elements. There is a 0% failure, and 56.82% of websites have passed this rule successfully. Providing preselected values

should be followed by the designers as a best practice to help users increase productivity with the application. Also, the data entered are already validated, thus processing a transaction also becomes much faster as server-side validations can be skipped. The user interface for mobile websites should maximize the use of radio buttons, selection lists and combo boxes in contrast to text boxes and text-area input controls. There should be a user-default pre-selected values wherever possible to minimize user input. Also, we need to make sure that empty controls are handled correctly.

Even though the rule of providing defaults has been followed by a number of designers, leading to a high passing percentage of 94%, still it can be improved by providing default values in every single case. Default values will surely lead to not having crashed pages on mobile devices by missing a page context or required parameter. The idea is to minimize user input as much as possible due to the limitations of using a small-screen device. The user faces a lot of difficulty typing with the small keypad on a mobile device. Sometimes, the user is unable to select the correct value in lists/combo boxes used on a webpage. In order to avoid the confusion and inconvenience caused to the user, default values that the user can utilize without much trouble should be used as much as possible.

CHAPTER 4. XML RULE SCHEMA

XML, also known as extensible markup language, has been used to create a rule schema. The rule schema defines all the W3C Mobile Web Guidelines that can be automated in the process of evaluating mobile websites. A set of rules has been represented in the form of rule schema and is used in the custom Mobile Websites evaluator. This set of rules is a subset of all rules used by the MobileOK checker in order to understand the semantics of the rule schema. The use of XML makes the rule schema flexible as well as extensible. The details about each rule is discussed in the rest of the chapter.

Semantics of the Rule XML Schema

Rule XML schema has the following hierarchy: on the topmost level, the *rule nodes* are present, followed by *tag nodes*; *attribute nodes*; and, finally, the *logical operator* and *comparison nodes*. The meaning and structure of all the nodes are explained later in the chapter.

Rule nodes

Each rule is represented by a rule node. A XML rule can have more than one *rule node*. A unique rule identifier is assigned in the rule node (`<rule id="1">`). Each of the child nodes for a rule node has a type attribute which is assumed to have a value equal to “*node*” or “*attribute*.”

Tag Nodes and Attribute Nodes

If the type equals “*node*,” it is called a *tag node*. The name of the *tag node* is used to search HTML tags for a mobile webpage. If the type equals “*attribute*,” it is called an *attribute node*. The name of the *attribute node* is used to search attributes of the HTML tags found by the parent *tag node*, which is explained more using the following example.

```
<rule id="1">
  <table type="node">
    <width type="attribute">
      <gr datatype="int">120</gr>
    </width>
  </table>
</rule>
```

In this schema, the *tag node* `<table>` represents the HTML tag `<table>` that needs to be checked on the mobile webpage. The child node for `<table>` is the *attribute node* `<width>`, which means that all width attributes for all the `<table>` tags of a mobile webpage are searched. The parent node of a *tag node* should be a *rule node*, or it can be a *tag node* itself. The parent node of an *attribute node* is always a *tag node*.

Comparison and Logical Nodes

The child node of the *attribute nodes* contains the logical definition of the rule. For ease of representation, four *comparison nodes*, `<gr>`, `<ne>`, `<eq>` and `<lt>`, have been used to represent the greater than, not equal, equal and less than operations. The comparison node can have an attribute named *datatype* which is used to perform the comparison. If *datatype* is not present, the rule evaluator assumes *datatype* to be string. The *comparison nodes* can also be enclosed in one of the following *logical operator nodes*: `<and>`, `<or>` or `<not>`. Each rule can have multiple conditions. Each condition needs to be evaluated individually to determine if the rule has passed or failed. If one of the

conditions is satisfied, the rule fails, and the automation tool moves to next rule in the rule schema. In this example, the *attribute node* `<width>` contains the `<gr>` *comparison node*. The value to be compared is nothing but the inner text of the *comparison node*.

Empty Attribute Nodes and Tag Nodes

An empty *tag node* implies that the existence of the corresponding HTML tag in the mobile webpage will violate the rule. Similarly, an empty *attribute node* implies that the attribute should not be present in the corresponding HTML tag of its parent *tag node*. If the tag is present, the rule is violated.

Rule XML Outline

A general outline of the Rule XML schema has been outlined in Figure 8 where the *rule nodes* are present at the topmost level, followed by *tag nodes*. The *tagname* in this XML structure can be replaced by the appropriate HTML tag that needs to be searched in the mobile webpage. The *attribute nodes* are the child node for any given *tag node*. The *attributename* can be replaced by the attribute of its corresponding parent *tag node* for which the rule is being validated. The *logical operator node* represents the `<and>`, `<or>` and `<not>` nodes, and it can be enclosed within the *tag nodes*. The logical operator nodes represent logical operations such as AND, OR and NOT. The *comparison nodes* are child nodes to *logical operator node* and can be replaced by any of the `<eq>`, `<ne>`, `<lt>` and `<gr>` nodes to specify the equal, greater-than, less than and greater than mathematical operations. The *datatype* of the *comparison node* needs to be specified with the values as integer, string, etc.

Figure 8. Rule XML Outline

```

<rule id="uniqueidentifier">
  <"tagname" type="node">
    .
    .
    .
    <"tagname" type="node">
      <"attributename" type="attribute">
        <"logicaloperatornode">
          <"comparisionnode"
datatype="typeofdata">value</"comparisionnode">
          .
          .
          .
          <"comparisionnode"
datatype="typeofdata">value</"comparisionnode">
        </"logicaloperatornode">
      </"attributename">
    </"tagname">
    .
    .
    .
  </"tagname">
</rule>

```

Keywords	Definitions
<i>uniqueidentifier</i>	Numeric identifier for a rule which is a unique v
<i>tagname</i>	HTML tag in a mobile webpage
<i>attributename</i>	Attribute of the corresponding HTML tag

Rules Implemented Using XML Rule Schema

Scrolling Should be Limited to One Direction

In order to test this rule, none of width attributes should be greater than the screen size for any standard mobile device. According to the default delivery context as defined in mobile W3C Mobile Web Best Practices, 120 pixels is the standard screen width considered for evaluating this rule [WMP08]. The XML representation of this rule is as follows:

```

<rule id="1">
  <body type="node">

```

```

    <width type="attribute">
      <gt datatype="int">120</gt>
    </width>
  </body>
  <table type="node">
    <width type="attribute">
      <gt datatype="int">120</gt>
    </width>
  </table>
  <tr type="node">
    <width type="attribute">
      <gt datatype="int">120</gt>
    </width>
  </tr>
  <td type="node">
    <width type="attribute">
      <gt datatype="int">120</gt>
    </width>
  </td>
</rule>

```

In this *rule node*, `<body>`, `<table>`, `<td>` and `<tr>` are the *tag nodes*. Each of these *tag nodes* has `<width>` as the *attribute node*. The *attribute nodes* have a *comparison node* as `<gt>` with *datatype* specified as an integer. The value to be compared is enclosed within the comparison nodes. In this case, the value is 120.

The *rule node* is comprehended in such a way that the mobile rule evaluator identifies four HTML tags, `<body>`, `<table>`, `<tr>` and `<td>`, corresponding to the *tag nodes*. The *rule node* then checks if the *width* attribute for any of these HTML tags has a value that is greater than 120 pixels on the mobile webpage. If the *width* attribute for any of these tags is greater than 120 pixels, the tool will return a failure.

Do Not Use Frames

This rule says that the mobile webpage should not make use of frames and other frameset elements, which means that the mobile webpage should not have a page within another page. In order to test this rule, the HTML page is searched for the presence of the

<frame>, <iframe> and <frameset> tags. The XML representation of this rule is as follows:

```
<rule id="2">
  <iframe type="node" />
  <frameset type="node" />
  <frame type="node" />
</rule>
```

The *rule node* has <iframe>, <frameset> and <frame> as the *tag nodes*. There are no *attribute nodes* present in the *rule node*, which implies that all three *tag nodes* are *empty tag nodes*. The mobile rule evaluator will check for the presence of the <frame>, <frameset> and <iframe> tags in the HTML webpage. The presence of any of these tags will violate the rule, and the tool will return a failure.

The Webpage Should Not Have Pop-ups

This rule says that there should not be any pop-up on mobile webpages. In order to test this rule, the mobile rule evaluator checks for the presence of the "target" attribute on all the links. The XML representation for this rule is as follows:

```
<rule id="4">
  <a type="node">
    <target type="attribute"></target>
    <and>
      <ne datatype="string">_self</ne>
      <ne datatype="string">_parent</ne>
      <ne datatype="string">_top</ne>
    </and>
  </a>
</rule>
</rule>
```

The *tag node* for this rule is the node <a> with <target> as the *attribute node*. The *attribute node* contains <and> as the *logical operator node* which contains <ne> as the *comparison node* with datatype equal to string. The *rule node* is used by the mobile rule evaluator to search for all <a> tags in the HTML page. If the value of the target attribute in

any of the `<a>` tags is not one of the following, "`_self`," "`parent`" or "`top`," the rule is violated.

Do Not Use Markup to Redirect Pages Automatically

This rule recommends that the mobile webpages should not be redirected automatically. Instead, the server should be configured to perform redirects by means of HTTP 3xx codes. The rule can be evaluated by checking the `http-equiv` attribute of the `<meta>` tag in any given mobile webpage, which should not be equal to "`refresh`." XML representation of the rule is as follows:

```
<rule id="5">
  <meta type="node">
    <http-equiv type="attribute">
      <eq>refresh</eq>
    </http-equiv>
    <content type="attribute"></content>
  </meta>
</rule>
```

The *tag node* for this rule node is the `<meta>` tag. The inner XML of the *tag node* contains the *attribute node* `<http-equiv>` and `<content>`. The `<http-equiv>` *tag node* consists of a *comparison node* `<eq>`, and the value enclosed that needs to be compared is equal to "`refresh`."

The mobile rule evaluator uses the *rule node* in such a way that the HTML tag `<meta>` is searched in the mobile webpage with the attribute `http-equiv` having a value equal to "`refresh`" or attribute `content` for the HTML tag `<meta>`. Presence of either of the above would defy the rule.

There Should be a Text Equivalent for Every Non-Text Element

This rule recommends using text for every non-text element on the mobile webpage because the webpages might have difficulty loading all the non-text elements on a small-

screen device due to a lower bandwidth or slow connection speed. In order to evaluate this rule, all image tags need to be searched within the HTML page with the *alt* attribute not equal to null or an empty string. If a value is present for the *alt* attribute for the image tag, the rule is passed; otherwise, the rule will return a failure. The XML representation of the rule is as follows:

```
<rule id="6">
  <img type="node">
    <alt type="attribute">
      <eq></eq>
    </alt>
  </img>
</rule>
```

To represent the above rule using XML, the *rule node* contains the *tag node* as ** with the *attribute node* as *<alt>* within the ** *tag node*. The *logical operator node* that is used in the *rule node* is *<eq>*. There is no value between the *<eq>* operators because the mobile rule evaluator compares with null or an empty string.

Do Not Create Periodically Auto-Refreshing Pages

This rule says that the mobile pages should not have the capability to refresh themselves periodically unless you have informed the user and provided a means of stopping it. In order to test this rule, the evaluator checks for the presence of a *<meta>* tag in an HTML mobile webpage having attributes *http-equiv* equal to *refresh* or *content* equal to the time limit.

This rule is evaluated using the Rule 5 rule node because we need to search for HTML tags with the same attribute and/or attribute values in order to determine whether the rule has been violated.

Check the Size of Images in the Markup if They Have an Intrinsic Size

This rule says that all images should specify the height and width attributes for a mobile webpage. XML representation of the rule is as follows:

```
<rule id="9">
  <img type="node">
    <height type="attribute"><eq></eq></height>
    <width type="attribute"><eq></eq></width>
  </img>
</rule>
```

The *rule node* consists of ** as the *tag node*, and *<height>* and *<width>* are the *attribute nodes*. The mobile rule evaluator uses this rule node to determine that there is no ** tag in the HTML page which does not have either the height or width values specified. In case where the value is not present, the tool returns a failure.

There Should be No Use of Nested Tables

The rule ensures that there are no tables that exist within another table in a mobile webpage. The XML representation of this rule is as follows:

```
<rule id="11">
  <table type="node">
    <table type="node">
    </table>
  </table>
</rule>
```

The *rule node* contains *<table>* as a *tag node* with another *tag node* nested within itself. There are no *attribute nodes* required to evaluate this rule. The inner *tag node* is an *empty tag node* implying the presence of a *<table>* tag within a *<table>* tag which violates the rule.

CHAPTER 5. AUTOMATION TOOL

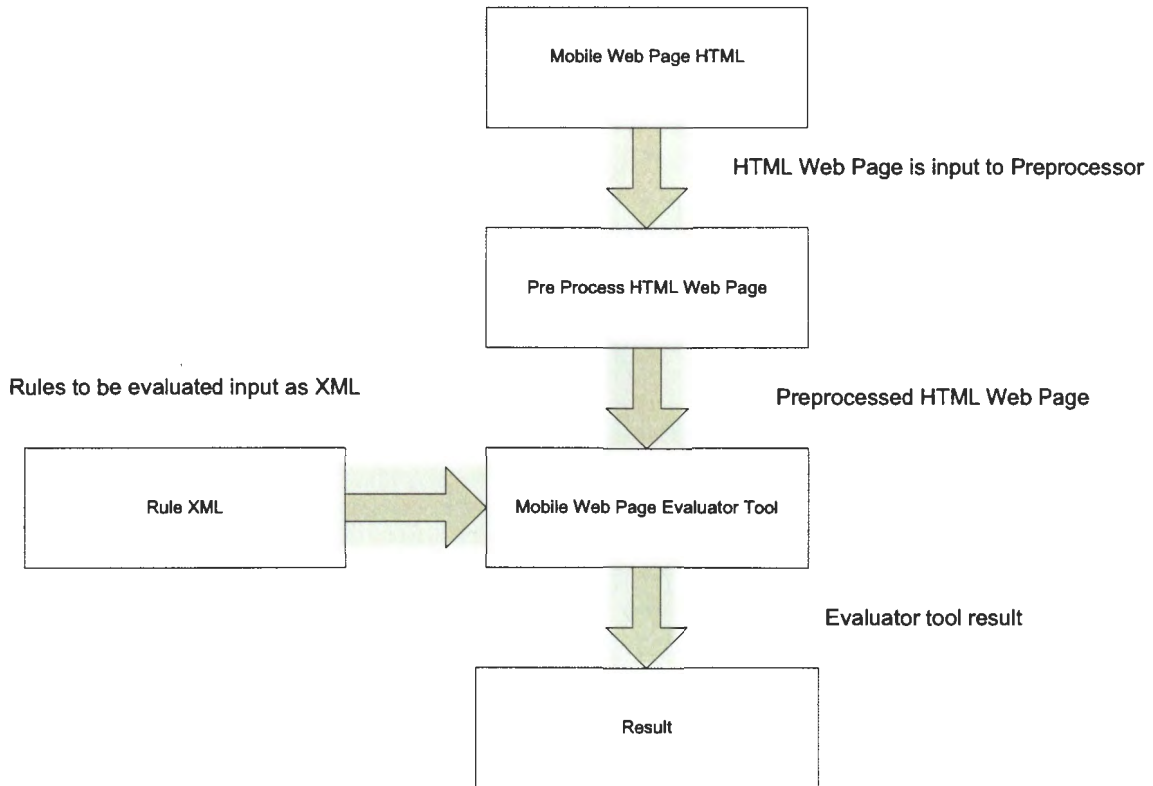
The automation tool is developed to evaluate a mobile webpage using the Rule XML. It takes the input as an HTML mobile webpage, and the rules are represented in the form of XML, known as Rule XML. The HTML webpage that is input to the tool is preprocessed prior to sending the input to the tool. Pre-processing the webpage ensures that any extraneous characters, comments, special characters, etc., are removed from the webpage. Also, pre-processing is done to remove any open tags present on the webpage and to insert quotation marks for the attribute values that are not enclosed within the quotes.

The processed webpage is first converted into XML. Because XML is a markup language, we need to make sure that all unwanted characters are removed from the webpage. Once the preprocessing is successful, the HTML page is converted into XML, and the rules are evaluated against the *rule nodes* taken from the Rule XML.

The tool evaluates the processed webpage against the Rule XML and outputs the result. The result states the rules passed or violated along with the reason for failure. A flowchart explaining the overall flow is given in Figure 9.

The classes that are used in the mobile rule evaluator tool are as follows: Pre-Processing HTML Page and Rule Evaluator. These classes are defined in more detail below.

Figure 9. Flow Chart for Evaluator Tool



Pre-Processing HTML Page

This class is used to pre-process the HTML webpage before it is input to the rule evaluator. Pre-processing is required because the HTML page needs to be converted into a valid XML structure. The pre-processing class includes the following functions.

Remove BR Tag

This function removes the BR tags that are present in the HTML mobile webpage. The removal of BR tags is essential because it is required to convert the HTML page into XML structure. The function searches for the presence of `
` and `</br>` tags, and replaces them with an empty string.

Remove NBSB

This function removes the non-breaking space or no-break space " " within a webpage used to provide a new line within a webpage, which is done so that the HTML page can be converted to a valid XML.

Remove Open Tags

This function removes unclosed tags, which means the tags that have a starting tag but do not have an ending tag. It accepts a string parameter called tag name. This check needs to be done because developers tend to forget to close some of the tags, i.e., any webpage that has a `<a>` but does not have its matching end tag, ``. The process to remove open tags includes the following steps:

1. Find the index of the first occurrence of the "`<[tagname]>`" string in the input HTML page, which is called *iStart*. The where tagname is the input parameter of the function.
2. Find the index of the next "`<[tagname]>`" character, which is called *iNext*.
3. If $iNext = -1$, then go to step 4; otherwise, go to step 5.
4. Find the closing bracket with a backslash `>`. If it is found, then the tag is already closed; if not, then a backslash is inserted before the closing bracket.
5. If the substring starting from position ($iNext + 1$) starts with `/[tagname]`, the tag is already closed. If not go to step 6; otherwise, go to step 7.
6. If the substring starting from *iStart* to ($iNext - 1$) contains `>`, the tag is a self-closed tag.
7. A tag is closed by inserting a backslash before the closing bracket.

8. Follow the same process, starting from step 1, with the substring between (*iNext*+2) to N.

Insert Quote for Attribute Values

This function inserts the quotation marks for all attribute values. For example, in webpages, the image width may be defined in the following way:

```
<img src="" width=50 height=20/>
```

This function will insert quotation marks for the width and height attribute values:

```
<img src="" width="50" height="20">
```

The algorithm for the function contains the following steps:

1. Find the index of the first occurrence of the "<" symbol in the input HTML page, which is called *iStart*.
2. Find the first occurrence of the ">" symbol, which is called *iEnd*.
3. Within the substring starting from the *iStart* index and going to *iEnd*, find all the occurrences of the "=" symbol.
4. For each occurrence, check if there is a double quotation mark after the "=" symbol. If yes, the value is already enclosed within the quotation marks; otherwise, insert quotation marks before and after the attribute value.
5. Repeat the steps 1 to 4 for the remaining HTML input string, starting from *iEnd*+1.

Rule Evaluator

This class is used to determine whether a rule has passed for any given HTML mobile webpage. It stores rule XML and the HTML page in the form of XML documents.

The rule evaluator class includes the following functions.

Get Rule

This function accepts a rule identifier as a string parameter defined in the rule XML and returns the *rule node* corresponding to that rule identifier.

Get Rule Description

This function accepts a rule identifier as a string parameter defined in the rule XML and returns the description of the rule which is present in the description attribute of the *rule node*. This description is used to show the result.

Evaluate Rule

This function accepts the rule identifier as a string parameter and returns information about whether the rule has been violated. If the rule is violated, the function also returns a brief message giving the reason for the failure. The algorithm for the function contains the following steps:

1. Find *rule node* corresponding to given rule identifier using the Get Rule function.
2. Find all the child nodes of the *rule node* and follow steps 3 to 10 for each child node.
3. If the node is a *tag node*, go to step 4; otherwise, continue with next child node.
4. If the inner text of the *tag node* is empty, then validate that the page does not have any tag corresponding to the *tag node*.
5. Otherwise, find all the child nodes of the *tag node*, and for each of the child nodes, follow steps 6 to 10.
6. If node is a *tag node*, then go to step 7; if the node is an *attribute node*, then go to step 8; otherwise, continue with the next child node.

7. Find all the HTML tags corresponding to the parent *tag node*, and validate that none of the tags has a child HTML tag corresponding to the current *tag node*.
8. If the inner text of the *attribute node* is empty, go to step 9; otherwise, go to step 10.
9. Find all the HTML tags corresponding to the parent *tag node*. Validate that none of these HTML tags has the attribute corresponding to the *attribute node*. If the attribute exists, the value of the attribute should be empty.
10. If the inner text of an attribute node is not empty, it should only be a logical *operator node*. If not, go to the next *attribute node*; otherwise, call the "Process Logical Operator Node" function.

Process Logical Operator Node

This function accepts the following parameters: a *logical operator node* as an XmlNode, the name of the HTML node upon which the rule needs to be validated, the attribute name from the Rule XML and the attribute value from the HTML webpage. It returns a message that is used as a reason for failure of the rule. The algorithm for the function contains the following steps:

1. If the name of the *logical node* is "and", evaluate each child *comparison node*, and perform an AND operation on the results obtained. To evaluate the child *comparison node*, go to step 3.
2. If the name of the *logical node* is "or", evaluate each of child *comparison node*, and perform an OR operation on the results obtained. To evaluate the child *comparison node*, go to step 3.

3. If the name of the node is "eq," check if the inner text of the *comparison node* is equal to the attribute value passed in the function. If they are equal, return true; otherwise, return false.
4. If the name of the node is "ne," check if the inner text of the *comparison node* is not equal to the attribute value passed in the function. If they are not equal, return true; otherwise, return false.
5. If the name of the node is "ge," check if the inner text of the *comparison node* is greater than the attribute value passed in the function. If it is greater, return true; otherwise, return false.
6. If the name of the node is "lt," check if the inner text of the *comparison node* is less than the attribute value passed in the function. If it is less, return true; otherwise, return false.

Note: *comparison nodes* have an attribute named "datatype" which is used to cast the inner text and an attribute value before doing the comparison.

Result Comparison with MobileOK Checker

Result comparison of the Automatic Website Evaluation Tool has been done against the MobileOK checker [WMC09] using various websites in order to find out if there are any discrepancy in the results of the Automatic Website Evaluation Tool. We find that the results are consistent in both applications, giving an overview that the results have been evaluated correctly by the automation tool. Also, there are some rules that are not evaluated by the MobileOK checker [WMC09], but they are part of the rule schema developed for the Automatic Website evaluation tool.

The analysis clearly shows that there are certain rules that provide results of the inconsistencies with the standard guidelines. The highest failure rates have been observed for Rule 3, Rule 5 and Rule 7, which indicate the use of large number of Pop-Ups, missing alternate text for the images, as well as missing height or width attributes in the `` tag used for a mobile webpage. The detailed analysis of the 46 websites performed as part of this paper also certifies the high failure rate of these guidelines and the need to educate web designers about these guidelines which should be followed for mobile webpage development.

For ease of understanding, we have listed the Rules in Table 1. The rules are the ones that have been used to represent the RuleXML schema in the automation Tool. The results of the website evaluation for the automation tool as well as the MobileOK checker are depicted in Table 2.

Table 1. Rules Used in Automation Tool

Rule Number	Rule Text
1	The scrolling should be limited to one direction, unless secondary scrolling cannot be avoided.
2	Do not use frames.
3	The webpage should not have pop-ups.
4	Do not use markup to redirect pages automatically.
5	There should be a text equivalent for every non-text element.
6	Do not create periodically auto-refreshing pages.
7	Check the size of images in the markup if they have an intrinsic size.
8	There should be no use of nested tables.
9	No image within a table

Table 2. Result Comparison of MobileOK Checker and Automation Tool

Website Name		Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8	Rule 9
Google	AT	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass
	MC	Pass	Pass	N/A	Pass	Pass	Pass	Pass	Pass	Pass
Paypal	AT	Pass	Pass	Fail	Pass	Fail	Pass	Fail	Fail	Pass
	MC	Pass	Pass	N/A	Pass	Fail	Pass	Fail	Fail	Pass
BankOfAmerica	AT	Pass	Fail	Fail	Pass	Pass	Pass	Fail	Pass	Pass
	MC	Pass	Fail	N/A	Pass	Pass	Pass	Fail	Pass	Pass
AmericanExpress	AT	Fail	Pass	Fail	Pass	Fail	Pass	Fail	Pass	Pass
	MC	Fail	Pass	N/A	Pass	Fail	Pass	Fail	Pass	Pass
eTrade	AT	Pass	Pass	Fail	Pass	Pass	Pass	Fail	Pass	Pass
	MC	Pass	Pass	N/A	Pass	Pass	Pass	Fail	Pass	Pass
ChaseMobile	AT	Pass	Pass	Fail	Pass	Fail	Pass	Pass	Pass	Fail
	MC	Pass	Pass	N/A	Pass	Fail	Pass	Pass	Pass	Fail
BusinessWeek	AT	Pass	Pass	Fail	Fail	Fail	Fail	Fail	Pass	Pass
	MC	Pass	Pass	N/A	Fail	Fail	Fail	Fail	Pass	Pass
Dictionarymobile	AT	Pass	Pass	Fail	Fail	Fail	Fail	Fail	Pass	Pass
	MC	Pass	Pass	N/A	Fail	Fail	Fail	Fail	Pass	Pass

CHAPTER 6. CONCLUSION AND FUTURE WORK

Conclusion

As part of the objective for the study, this paper performed a detailed analysis of the 46 most popular mobile websites against the MobileOK checker. The MobileOK checker is a free service provided by W3C that can be used to check the mobile friendliness of any website. The analysis is performed categorically by grouping the guidelines into different categories, such as Navigation and Links, Page Layout and Content, Page Definition, Image Size, Tables, Style Sheets, and User Input.

The analysis provides an in-depth understanding of the failure rates for the guidelines within each of the categories as well as which guidelines have overall high failure rates. Among the Navigation and Links category, the guideline that is violated the most is “external resources.” Navigation and Links is one of the most vital category because minimizing the use of external resources, such as images, style sheets as well as other additional links, for any website will decrease the overall load time of the website on a mobile device. This guideline also overlaps with a number of other guidelines that convey a similar message.

For the Page Layout and Content category, the rule that has been violated the most is the “Page Size” rule. The page size has a limit of 20 KB according to W3C [WMP08]. To make sure that the page size remains within the limit, developers can avoid the use of resources, CSS, scripts, etc.

Under the Page Definition category, “Valid MarkUp,” “Content format support” and “Character Encoding support” are the rules that have a high failure rate and require

attention from web developers. The importance of these rules needs to be understood and applied in the real world. It is extremely important to realize that a failure to follow these guidelines can result in failure to render these websites correctly on a mobile device.

The next category focuses strictly on image sizes. Analysis shows that not many designers follow the “Image Specify size” rule. The image size should be specified, which helps the mobile browser load an image correctly. “Image Specify size” rule should be followed as an industry-wide standard.

The Tables category stresses the importance of a contemporary table-less design and an alternate way of making use of a `<div>` tag to create websites instead of the traditional use of strict table design. The table-less design gains importance because it provides more flexibility that is required for creating websites for small-screen devices. The “tables layout” rule has been violated the most, which means that website designers still create websites using the tables. The new approach of a table-less design is not very familiar, and efforts need to be taken to create awareness about it.

The style sheets category focuses on the use of style sheets instead of using the embedded style attributes. Most websites fail the “style sheets use” guideline, which means that some style components are part of the webpage itself. Embedding the style components should be avoided as much as possible because it makes the pages heavier as well. The external CSS can be cached on the first webpage request, and subsequent requests can use the cached style sheet.

Last but not least, under the User input category, none of the rules has a failure rate, although “default input mode” has a high warning rate. A number of designers are following this rule where preselected values are provided wherever possible on the

webpages so that minimal user involvement and inputs are required, thus saving users' time and energy.

As we see, most guidelines focus on one common goal to enhance the usability of mobile webpages as well as providing an unproblematic experience of accessing websites from a mobile phone or any other small-screen device. The analysis also provides insights about understanding the common mistakes made by the website developer and how the mistakes can be avoided by remembering the results of this analysis.

To achieve the second objective of the study, the paper also presents a format for representing the W3C guidelines that makes use of an open format, XML. The format is very easy to understand and provides flexibility for changing an already-existing rule. It also provides the ease of extending the rule base to include new rules which developers might want to add in the course of testing a work-in-process website.

Furthermore, the research also has an objective to develop a tool that uses the above-mentioned XML format to perform automatic analysis of mobile websites. The tool was built using Object Oriented Design and has been able to successfully evaluate mobile webpages and to provide results about inconsistencies with the standard guidelines.

Future Work

In this paper, we concentrated on a subset of all the guidelines provided by W3C that can be represented in XML format. The representation of all the guidelines in the proposed XML format is a challenging task, and more research can be done in this area. Moreover, this research can also be extended to achieve 100% automatic evaluation of mobile websites. Currently, the created tool uses some guidelines and provides a direction to move forward towards achieving total automation.

We have also not considered the different physical properties of various mobile devices. Most mobile devices have different screen sizes and abilities to display web content in a different manner. Due to these differences, mobile webpages could be rendered in different ways on these devices. More research can be done in this field where a mobile device's physical properties can be taken into consideration during the evaluation process. The usefulness of the research can be realized by considering different mobile devices commonly used by the people all over the world.

The XML rule schema proposed in the paper can be further elaborated so that more guidelines defined in W3C can be represented in that format. Increased number of guidelines will enhance the use of a tool that has an open standard and is also independent of the guidelines because a set of guidelines are provided separately as input to the developed guideline checker.

REFERENCES

- [Arr06] M. Arrue, M. Vigo and J. Abascal, "Automatic Evaluation of Mobile Web Accessibility," *Universal Access in Ambient Intelligence Environments*, pp. 244-260, 2006.
- [Ber04] E. Bertini and G. Santucci, "Modeling Internet Based Applications for Designing, Multi-Device Adaptive Interfaces AVI," *Proceedings of the Working Conference on Advanced Visual Interfaces*, Gallipoli, Italy, pp. 252-256, 2004.
- [Bjo00] S. Bjork, L. E. Holmquist, P. Ljungstrand and J. Redstrom, "PowerView: Structured Access to Integrated Information on Small Screens," *Conference on Human Factors in Computing Systems CHI '00*, The Hague, The Netherlands, pp. 265-266, 2000.
- [Bis06] J. Bishop, "Multi-Platform User Interface Construction—A Challenge for Software Engineering-in-the-Small," *Proceedings of the 28th International Conference on Software Engineering*, Shanghai, China, pp. 751-760, 2006.
- [Bor96] J. A. Borges, I. Morales and N. J. Rodriguez, "Guidelines for Designing Usable World Wide Web Pages," *ACM CHI 96 Conference on Human Factors in Computing Systems*, Vol. 2, pp. 277-278, 1996.
- [Bor07] Y. Borodin, J. Mahmud and I. V. Ramakrishnan, "Context Browsing with Mobiles-When Less is More," *International Conference on Mobile Systems, Applications and Services*, *Proceedings of the 5th International Conference on Mobile Systems*, Juan, Puerto Rico, pp. 3-15, 2007.
- [Bro88] C. M. Brown, "Human-Computer Interface Design Guidelines," Ablex Publishing Corporation, Norwood, NJ, USA, 1988.

- [Can08] "Top Mobile Websites for 2008," URL: <http://www.cantoni.org/2009/01/01/top-mobile-sites-2008>, 2008.
- [Car05] A. Carrillo-Ramos, J. Gensel, M. Villanova-Oliver and H. Martin, "PUMAS: A Framework Based on Ubiquitous Agents for Accessing Web Information Systems Through Mobile Devices," ACM Symposium on Applied Computing, Santa Fe, New Mexico, pp. 1003-1008, 2005.
- [Chu07] K. Church, B. Smyth, P. Cotter and K. Bradley, "Mobile Information Access: A Study of Emerging Search Behavior on the Mobile Internet," ACM Transactions on the Web, 1(1), 2007.
- [Des06] L. G. DeSousa and J. C. Leite, "Using IMML and XICL Components to Develop Multi-Device Web-Based User Interfaces," ACM International Conference Proceeding Series, Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems, Natal, Brazil, pp. 138-147, 2006.
- [Des90] F. deSouza and N. Bevan, "The Use of Guidelines in Menu Interface Design: Evaluation of a Draft Standard," Human-Computer Interaction, 435-440, 1990.
- [Dix98] A. Dix, J. Finlay, G. Abowd and R. Beale, "Human-Computer Interaction" (second edition), Upper Saddle River, NJ, Prentice Hall, 1998.
- [Eis01] J. Eisenstein, J. Vanderdonckt and A. Puerta, "Applying Model-Based Techniques to the Development of UIs for Mobile Computers," International Conference on Intelligent User Interfaces Proceedings of the 6th International Conference on Intelligent User Interfaces, Santa Fe, New Mexico, pp. 69-76, 2001.
- [Fre01] J. Freire, B. Kumar and D. Lieuwen, "WebViews: Accessing Personalized Web Content and Services," International World Wide Web Conference Proceedings of the

- 10th International Conference on World Wide Web. Hong Kong, Hong Kong, pp. 576-586, 2001.
- [Jef91] R. Jeffries, J. R. Miller, C. Wharton and K. M. Uyeda, "Software and Systems Laboratory," HPL-91-03, January 1991.
- [Gon02] F. J. Gonz'alez-Casta~no, J. Vales-Alonso, M. Livny, E. Costa-Montenegro and L. A. Condor, "Grid Computing from Mobile Handheld Devices," ACM SIGMOBILE Mobile Computing and Communications Review, 6(2), pp. 18-27, 2002.
- [Gru03] J. Grundy and B. Yang, "An Environment for Developing Adaptive, Multi-Device User Interfaces," ACM International Conference Proceeding Series, Proceedings of the Fourth Australasian User Interface Conference on User Interfaces, Vol. 18, pp. 47-56, 2003.
- [Hal05] B. J. Halpert, "Authentication Interface Evaluation and Design for Mobile Devices," Proceedings of the 2nd Annual Conference on Information Security Curriculum Development, Kennesaw, GA, pp. 112-117, 2005.
- [Ivo00] M. Y. Ivory and M. A. Hearst, "Automated Usability Evaluation of WIMP and Web Interfaces," CHI 2000 Conference.
- [Ivo01] M. Y. Ivory and M. A. Hearst, "The State of the Art in Automating Usability Evaluation of User Interfaces," ACM Computing Surveys, 33(4), pp. 470-516, 2001.
- [Kai03] A. Kaikkonen and V. Roto, "Navigating in a Mobile XHTML Application," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Ft. Lauderdale, Florida, pp. 329-336, 2003.
- [Kiy01] H. Kiyomitsu, A. Takeuchi and K. Tanaka, "ActiveWeb: XML-Based Active Rules for Web View Derivations and Access Control," Proceedings of the Workshop

on Information Technology for Virtual Enterprises, Queensland, Australia, Vol. 13, pp. 31-39, 2001.

[Lam05] 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, Portland, Oregon, pp. 681-690, 2005.

[Lec98] A. Lecerof and F. Paternò, "Automatic Support for Usability Evaluation," IEEE Transactions on Software Engineering, 24(10), pp. 863-888, 1998.

[Lee03] K. B. Lee and R. A. Grice, "An Adaptive Viewing Application for the Web on Personal Digital Assistants," ACM Special Interest Group for Design of Communication," Proceedings of the 21st Annual International Conference on Documentation, San Francisco, CA, pp. 125-132, 2003.

[Lee04] Y. E. Lee and I. Benbasat, "A Framework for the Study of Customer Interface Design for Mobile Commerce," International Journal of Electronic Commerce, 8(3), pp. 79-102, 2004.

[Leh06] T. Lehtonen, S. Benamar, V. Laamanen, I. Luoma, O. Ruotsalainen, J. Salonen and T. Mikkonen, "Towards User-Friendly Mobile Browsing," ACM International Conference Proceeding Series," Proceedings of the 2nd International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications, Pisa, Italy, 198(6), 2006.

[Lew90a] C. Lewis and P. Polson, "Theory-Based Design for Easily-Learned Interfaces," Human-Computer Interaction, 5, 191-220, 1990.

- [Lew90b] C. Lewis, P. Polson, C. Wharton and J. Rieman, "Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces," Proceedings of CHI'90, Seattle, Washington, ACM, pp. 235-242, 1990.
- [Low92] J. Lowgren and T. Nordqvist, "Knowledge-Based Evaluation as Design Support for Graphical User Interfaces," CHI'92 Conference on Human Factors in Computing Systems, ACM, pp. 181-188, 1992.
- [Nie93] J. Nielsen, "Usability Engineering," Boston, MA, Academic Press, 1993.
- [Nie00] J. Nielsen, "Designing Web Usability: The Practice of Simplicity," Thousand Oaks, CA, New Riders Publishing, 2000.
- [Nie90] J. Nielsen and R. Molich, "Heuristic Evaluation of User Interfaces," Proceedings of CHI'90, ACM, pp. 249-256, 1990.
- [Obr07] Z. Obrenovi' and J. V. Ossenbruggen, "Web Browser Accessibility Using Open Source Software." ACM International Conference Proceeding Series, Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A) Banff, Canada, Vol. 225, pp. 15-24, 2007.
- [Rat96] J. Ratner, E. M. Grose and C. Forsythe, "Characterization and Assessment of HTML Style Guides," ACM CHI'96 Conference on Human Factors in Computing Systems, Vol 2, pp. 115-116, 1996.
- [Sal05] A. Salzburg, "Serendipitous File Exchange Between Users and Devices," ACM International Conference Proceeding Series, Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services, Vol. 111, pp. 39-46, 2005.

- [Sch98] J.Scholtz, S. Laskowski & L.Downey, "Developing Usability Tools and Techniques for Designing and Testing Web Sites", In Proceedings of the 4th Conference on Human Factors & the Web, 1998.
- [Shn86] B. Shneiderman, "Designing the User Interface: Strategies for Effective Humancomputer interaction," Boston, MA, Addison-Wesley, 1986.
- [Shn00] B. Shneiderman, "Universal Usability," Communications of the ACM, 43(5), 84-91, 2000.
- [Shn03] B. Shneiderman, "Promoting Universal Usability with Multi-Layer Interface Design," ACM Conference on Universal Usability Proceedings of the 2003 Conference on Universal Usability, Vancouver, British Columbia, Canada, pp. 1-8, 2003.
- [Smi86] S. L. Smith, "Standards versus Guidelines for Designing User Interface Software, Behaviour and Information Technology", 47-61, 1986.
- [Sun07] Z. Sun, J. Mahmud and I. V. Ramakrishnan, "Model-Directed Web Transactions Under Constrained Modalities," ACM Transactions on the Web, 1(3), 2007.
- [Vig08] M. Vigo, A. Aizpurua, M. Arrue and J. Abascal, "Evaluating Web Accessibility for Specific Mobile Devices," Proceedings of the 2008 International Cross-Disciplinary Conference on Web Accessibility (W4A), pp. 65-72, 2008.
- [WCAG99] W3C, Web Content Accessibility Guidelines 1.0, 1999.
- [WMC09] W3C, MobileOK Checker, URL: <http://validator.w3.org/mobile>, 2009.
- [WMP08] W3C, Mobile Web Best Practices 1.0, URL: <http://www.w3.org/TR/mobile-bp>, 2008.
- [WMT08] W3C, MobileOK Basic Tests 1.0, URL: <http://www.w3.org/TR/mobileOK-basic10-tests> , 2008.

[WMT09] 7 Media types, W3C Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification, URL: <http://www.w3.org/TR/2009/CR-CSS2-20090908>, 2009.

[Zha98] Z. Zhang, V. Basili and B. Shneiderman. "An Empirical Study of Perspective Based Usability Inspection," Proceedings of the Human Factors and Ergonomics," 42nd Annual Meeting of the Human Factors Ergonomics Society, pp. 1346-1350, 1998.