COMPUTER
ENGINEERING
PROGRAM

Middle East Technical
University
Northern Cyprus
Campus

eMINE Technical Report Deliverable 7 (D7),

October 2013

Web Page Transcoding Based on Eye Tracking

M. Elgin Akpınar
elgin.akpinar@metu.edu.tr
Middle East Technical University,
Ankara, Turkey

Yeliz Yesilada
yyeliz@metu.edu.tr
Middle East Technical University
Northern Cyprus Campus,
Kalkanlı, Güzelyurt, TRNC,
Mersin 10, Turkey

As web technologies evolve and enable designers to create more visually interactive and consequently complex web pages, unfortunately, accessing these pages in alternative forms such as by using screen readers or mobile devices, is still a major problem. In the context of eMINE project, an algorithm to visually identify visual elements and their roles in the web pages, and eMINE Scanpath Analysis Algorithm to generate a common scanpath for visual elements by using eye tracking data were proposed. In this technical report, a number of transcoding techniques and their architectures are discussed. These techniques aim to combine the visual element identification with eMINE Scanpath Analysis Algorithm.

METU
NCC

eMINE

The World Wide Web (web) has moved from the Desktop and now is ubiquitous. It can be accessed by a small device while the user is mobile or it can be accessed in audio if the user cannot see the content, for instance visually disabled users who use screen readers. However, since web pages are mainly designed for visual interaction; it is almost impossible to access them in alternative forms. Our overarching goal is to improve the user experience in such constrained environments by using a novel application of eye tracking technology. In brief, by relating scanpaths to the underlying source code of web pages, we aim to transcode web pages such that they are easier to access in constrained environments.

Acknowledgements

The project is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) with the grant number 109E251. As such the authors would like to thank to (TÜBİTAK) for their continued support.

Contents

Middle East Technical University,
Northern Cyprus Campus,
Kalkanlı, Güzelyurt, TRNC,
Mersin 10, TURKEY

Tel: +90 (392) 661 2000
http://www.ncc.metu.edu.tr/

Corresponding author:
M. Elgin Akpınar
elgin.akpinar@metu.edu.tr

1 Introduction

With improvement of web and its technologies, design methods and processes has changed. Increase of visual interaction in web pages resulted in more sophisticated layouts and web page structures. Unfortunately, accessibility remains as one of the major problems to web. Although web pages provide a rich way of visual interaction with visual elements, these visual elements are not available when pages are accessed in alternative forms, such as in audio with assistive technologies.

There are many assistive technologies that support accessibility of web pages to disabled people which include screen readers, screen magnifiers, etc. (Edwards, 2008). Screen readers typically read the underlying source code of the page by starting from the first node of the DOM and following the node order. Therefore, this sequential access makes browsing time-consuming and strenuous for screen reader users (Borodin et al, 2007; Mahmud et al, 2007a,b). When the web page includes a header which contains many links or a left side bar which consists of many subcontents, the time required to access the main content increases. For example, Figure 1 shows a typical web page. This page is very easy for an ordinary user to access the important part of the page in a few seconds. However, when we tried to access these page by using a screen reader, it read all of the items in header and left side bar, which took around 40 seconds and indeed is not a reasonable amount of time for everyday use of internet. In order to solve this problem, we proposed a visual element identification algorithm, so that, screen reader may jump to the visual element, which user may be interested in, immediately. Moreover, the web pages, which are designed for visual

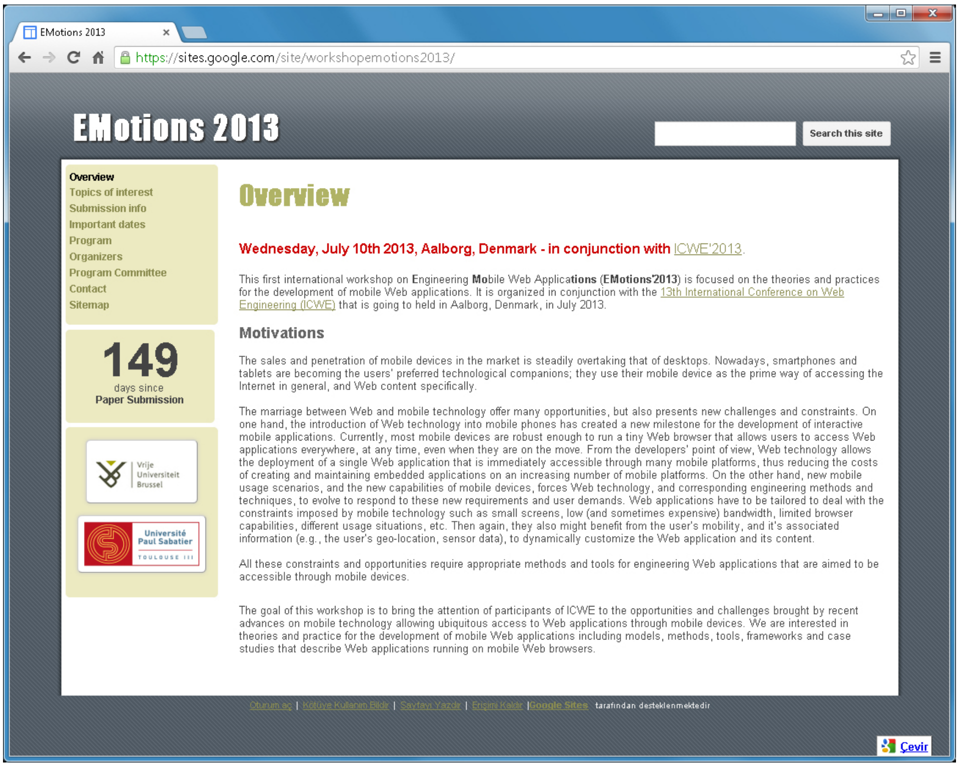


Figure 1: Emotions 2013 Home Page

interaction, include many items encoded in visual rendering of the page, and these pages become inaccessible in screen readers.

Accessing web content has also some difficulties in mobile devices, due to some limited capabilities and resources of these devices (Hattori et al, 2007; Yin and Lee, 2004; Zakas, 2013). One major problem is the screen size. Although screen sizes are getting larger, they are still very small when they are compared to desktop screen sizes. Therefore, navigating in a web page, which originally designed only for large screens, is an important problem (Ahmadi and Kong, 2012; Milic-Frayling and Sommerer, 2002). Another problem is limited memory of device and bandwidth. Especially for large pages, it is a costly procedure to download the content of the page and process it (Xie et al, 2005; Zakas, 2013). Moreover, modern devices typically do not provide a keyboard or mouse, rather they are used with touch screens. Therefore, some aspects of keyboard and mouse interaction in desktop are not available in mobile devices (Chen et al, 2009; Xiao et al, 2005; Xie et al, 2005). There are some frameworks, such as Bootstrap[1], which were designed to solve such problems, unfortunately, they are used rarely by the designers. Furthermore, according to our observation, not every web page also provides a mobile mirror. These drawbacks of mobile web access arise the requirement of an alternative representation for mobile devices (Adzic et al, 2011).

This technical report first discusses a number of transcoding strategies, and then, focus on our eye tracking based web page transcoding technique. Based on the common scanpaths generated with eMINE Scanpath Analysis Algorithm (Eraslan et al, 2013), first of all we define a role path by using our visual element identification algorithm. Then, these role paths are used to navigate user through the page. In order to solve the accessibility problems of both screen readers and mobile devices, we propose a proxy based system, which segments a web page and returns only the instance of current role in the role path. Users navigate between visual elements by using some interaction items and according to the type of interaction a new visual element is returned to the user.

The rest of this technical report has been organized in the following way: Section 2 explains our visual element identification algorithm and eMINE Scanpath Analysis Algorithm which were developed in the context of eMINE Project. Section 3 covers recent studies on web page transcoding to solve web accessibility and mobile device adaptation problems. Section 4 discusses some transcoding methods to solve the accessibility problem. Then, Section 5 describes our proxy side architecture for web page transcoding. Finally, Section 6 discusses future work and the alternative delivery models for transcoding, and Section 7 concludes our technical report.

2 Background Work

The transcoding method described in this technical report is based on two algorithms: visual element identification algorithm and eMINE Scanpath Analysis Algorithm.

[1] http://getbootstrap.com/

2.1 Visual Element Identification Algorithm

Visual element identification algorithm aims to identify visual elements in a web page by using some heuristics. First of all, it segments a web page by using an extended version of VIPS Algorithm to detect meaningful visual elements in a web page (Akpınar and Yeşilada, 2012). Then, it detects the roles of each element by applying some heuristic rules on them. The heuristic rules are defined in an ontology based knowledge base and rules are applied in a rule engine. At the end of its process, the algorithm labels each visual element with its corresponding role in the web page (Akpınar and Yeşilada, 2013a,b).

2.2 eMINE Scanpath Analysis Algorithm

eMINE Scanpath Analysis Algorithm aims to generate a common scanpath by using eye tracking data and the result tree of web page segmentation process. It projects the fixations of eye tracking data over the visual elements and generates a path for each user. Then, the algorithm calculates the most common scanpath by using these individual paths. First of all, it finds two most similar scanpaths by using Levenshtein Distance, and then finds the common scanpath by applying Longest Common Subsequence method over these two scanpaths. It iterates until only one scanpath is left in the list (Eraslan et al, 2013; Yesilada et al, 2013).

3 Related Work

Mobile device adaptation and web accessibility are two major areas in which there are some accessibility problems. There are many researches which propose transcoding methods to solve such problems. In this section, we summarize these researches. However, none of these techniques takes neither the heuristic roles of visual elements in detail, nor the user behavior with eye tracking experiments.

3.1 Small Screen Adaptation

In recent years, there has been an increasing amount of literature on small screen adaptation of web pages. One of the small screen adaptation methods is to remove unnecessary or irrelevant content from the web page. Xiao et al (2008a) first segment a web page, and then remove irrelevant parts of the page. Similar to Xiao et al (2008a), Chen et al (2001) label some visual elements as decoration and special objects, such as advertisement objects and remove these visual elements from the layout. However, removing such objects does not ensure that the remaining content fits in the screen. Moreover, users may want to access to advertisements on purpose, then some specific parts of the web page becomes inaccessible. Some researches, on the other hand, aims to display a specific part of the web page to the user. Xie et al (2005); Yin and Lee (2005) first identify the most important part of the web page, and then display this part to the user. Similarly, Yin and Lee (2004) aim to extract and present only the important parts of the web page. However, the visual elements which are identified as unimportant may be very important for a set of user, and become inaccessible for such users. Unlike Yin and Lee (2004) and Xie et al (2005); Yin and Lee (2005), Song et al (2004) sort the visual elements in the order of their importance and provide a sequential access to visual elements.

Another possible technique is to summarize the web page and provide a specific part of the page based on the users' selections. Ahmadi and Kong (2008); Hattori et al (2007) divide the web pages into smaller subpages, and create a table of content of the overall web page. Chen et al (2003, 2005) also divide the web pages into smaller subpages and connect these subpages by with each other. Similarly, Hwang et al (2003) segment a web page into smaller subpages and bind subpages by providing some hyperlinks in a sequential order. Summarization can be achieved with alternative representation of visual elements. Xiao et al (2005) present the overall page to the user and provide a model, which is called tap and display model, and provides the content of the subpage that user selected. Baluja (2006); Chen et al (2003, 2005); Xiang et al (2007); Xiao et al (2008b); Yang and Shi (2009) propose a thumbnail view in which, the web page is summarized to thumbnails and users access to the content by selecting the thumbnails. Xie et al (2005) take this approach further by providing an importance highlighting method. Some researches, on the other hand, aim to summarize the visual elements, rather than summarizing the web page. Hwang et al (2003); Whang et al (2001) keep the original layout and convert each header to a link, which points to the content of the particular visual element. Another summarization technique is to remove the content of each segment except for the first sentences of those segments (Ahmadi and Kong, 2008).

Some web pages contain media objects such as an embedded video. If not specified by the designer, the object may not fit in the screen. These visual elements can be scaled down to fit in the small screen (Adzic et al, 2011). Another proposed method is modality conversion. Changing the presentation of modality of an object can solve the representation problem (Adzic et al, 2011). However, in this project we are not interested in modality conversion, we aim to display visual elements in their original format.

3.2 Web Accessibility

A considerable amount of literature has been published on web accessibility. Asakawa and Takagi (2000); Takagi et al (2002) aim to make web pages easily accessible by screen readers, by identifying segments to reengineer the web page. Their method consists of manual annotation of each segment with a predefined set of roles, such as header, footer, etc. Then, the page is transcoded with respect to the role of each segment. In this transcoding, more important visual elements are moved to the top of the page, while less important segments are moved to the bottom of the page. Moreover, Yesilada et al (2007) aim to describe the roles of segments with a rich vocabulary, which is encoded in an ontology called WAfA, and then transcode web pages to support better accessibility. Similarly, Harper et al (2006); Lunn et al (2008a,b) propose an annotation method, in which, they annotate the CSS attributes with a simpler set of vocabulary. However, these approaches are based on manual annotation, rather than automatically discovering of visual elements.

Another study by Borodin et al (2007); Mahmud et al (2007a,b) indicate that, assistive applications such as screen readers process a web page sequentially with respect to the DOM structure. Consequently, browsing becomes time-consuming and strenuous for screen reader users. In order to solve this problem, they propose a method which implements a geometrical clustering algorithm to identify segments in a web page automatically, skips some unnecessary parts and jumps to the main content.

4 Transcoding Strategies

In this section, we discuss basic transcoding strategies which we deploy in our architecture. These strategies mainly address the accessibility problems for either screen readers, mobile devices or both. The first strategy is selective representation, which aims to represent a single visual element at a time. The second strategy is summarization of the web page, based on the roles of visual elements. Another strategy is sequential and direct access to the visual elements with respect to the user interaction with the client application. Other strategies are related with the modification of visual elements in order to fit them into the screen. All of these strategies are handled at the proxy side of the application. Selective representation and sequential access strategies benefit the outcomes of eye tracking studies, while the others are related with the page structure and screen size of the device.

4.1 Selective Representation

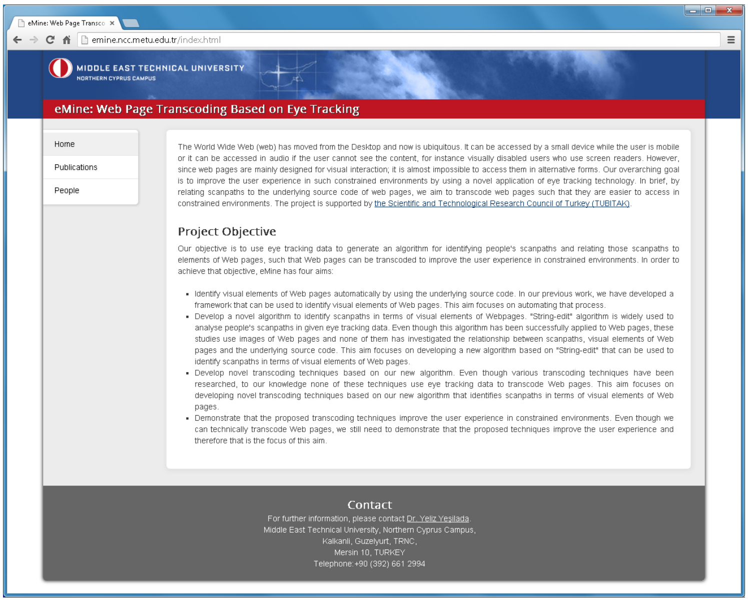
Web pages consist of many visual elements and user may not be interested in some of these elements. In screen readers, this would increase the time required for a user to access a visual element which the user is interested in, since screen readers typically follows the order in which the visual elements are placed in DOM structure. In mobile devices, these visual elements cause unnecessary use of limited resources, such as bandwidth, as well as producing a complex representation in the small sized screen.

In order to solve this problem, a typical approach is to select a specific visual element at each time and represent it to the user. This approach would simplify the page content and provide simplistic view in small screen devices and reduce the time to reach important content. Our approach is similar to the visual element removal methods in the literature, expect for the fact that, current visual element is selected based on eye tracking study results, rather than following the order of DOM structure. Figure 2 illustrates the effect of this operation in a web page. By selecting only a specific part (in this case, the main content), the web page in Figure 2a is represented by removing unnecessary parts as in Figure 2b.

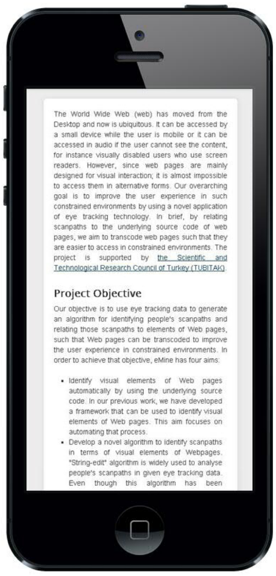
4.2 Layout Modification

Typical web page layout is in the form of a grid[2], where designers both uses the aspects of larger width and height of the screen. However, when it comes to a mobile device, all of the content cannot be represented in the screen at once and user requires to navigate both horizontally and vertically. This, indeed, makes navigation harder. In order to reduce navigation into one direction, the row structured content may be split into columns and the row may be reorganized, so that, the columns are placed in vertical layout. For example in eMINE page (Figure 3a), the row of people is originally horizontal. However, it becomes vertical in small screens to provide a better view (Figure 3b).

[2] `http://www.w3.org/TR/css3-grid-layout/`

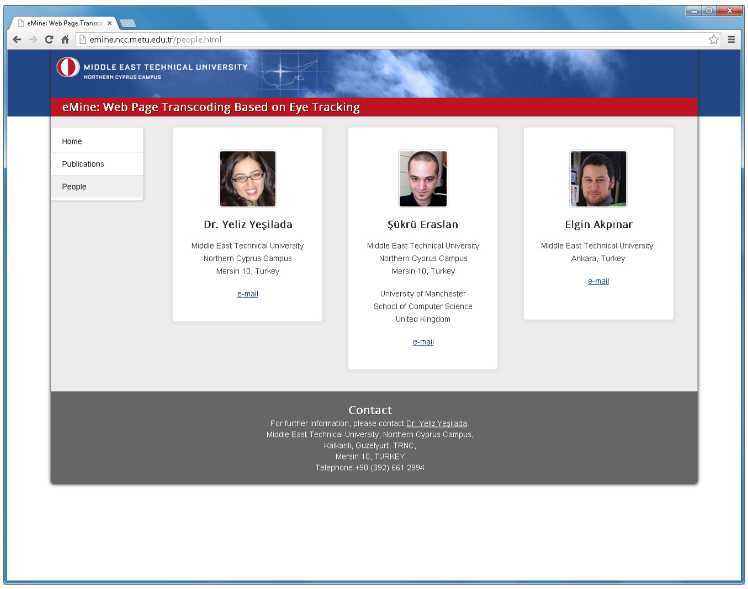


(a) Desktop view

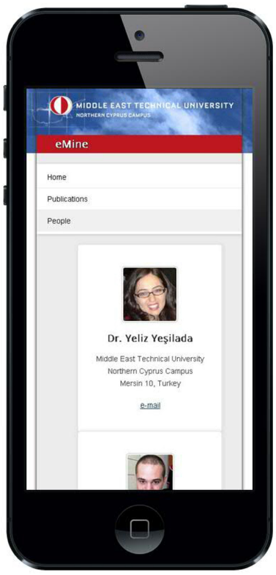


(b) Mobile view

Figure 2: eMINE - Home page in both desktop and mobile



(a) Desktop view



(b) Mobile view

Figure 3: eMINE - People page in both desktop and mobile

4.3 Representational Manipulation

One main problem which we have observed in current web page designs, is that many visual elements have constant widths. Although this may seem useful in providing a better representation in desktop browsers, it causes some visual problems in small screen devices, since the screen size is smaller than the width of the visual elements. Therefore, users have to navigate both vertically and horizontally in the web pages, consequently it makes harder to navigate in the web pages. Also, some media elements such as images may not fit in the screen in case of constant size definitions. One possible solution for such cases is to assign relative width values rather than constant values. Also, it is possible to provide different representational styles by using CSS Media Queries[3]. Unfortunately, many designers do not use these techniques in their web pages. In order to solve this problem, we need to make visual elements fit in the screen by modifying the underlying source code and apply relative size and media query techniques on visual element. Figure 2 illustrates a good example of such cases. The main content in Figure 2a is nicely fit in the screen as in Figure 2b, so that, user can read the content easily.

4.4 Summarization

Current studies show that, providing a summary of a web page is very useful. In these studies, the summary of the page is either in the form of table of contents (Ahmadi and Kong, 2008; Hattori et al, 2007) or each visual elements are represented as thumbnails (Baluja, 2006; Chen et al, 2003, 2005; Xiang et al, 2007; Xiao et al, 2008b; Yang and Shi, 2009). Since thumbnail representation may cause to accessibility issues for disabled users, we deploy a summarization method which is similar to table of content approach. The difference in our approach is that, the summary of the page only consist of the role classes which appear in the page. Furthermore, since a web page may contain many instances of a single role, it is important to provide a further summary of a selected role. However, all of the instances are in the same role, therefore, either title or the first sentence of the visual element should be used in this second summary. Summaries should be accessible with a certain interaction method and provide direct access to the visual elements.

4.5 Sequential and Direct Access

Our main purpose in this method is to provide sequential access to the visual elements, which is based on a path generated from the eye tracking study. Using this path, users may navigate backward or forward with certain types of interaction. The idea is very similar to the existing methods in the literature, except that it takes the path from eye tracking results into consideration. However, knowledge base for heuristic roles of visual elements in visual element identification algorithm contains some important roles, which users may want to access directly, such as menu. In order to provide this functionality, user interaction of the application should reserve some interaction types for these important roles to access directly. For example, after pressing to ’m’ in keyboard, user may jump to menu visual element immediately. Also, there should be additional keywords to be able to jump at the beginning or at the end of the path.

[3] `http://www.w3.org/TR/css3-mediaqueries/`

5 Web Page Transcoding Based on Eye Tracking

In this section, we construct a knowledge base for common scanpaths, define the basic user interaction methods and describe the overall system architecture. Figure 4 represents the proxy architecture in detail. It consists of the visual element identifier (1) and a knowledge base for common scanpaths (2). Also, it includes a visual element tree cache (3), which is constructed by using 1 and 2, to enable users navigate from a state (4) to another state (5) based on their interactions with the system. The representation of the visual elements in each state is determined with respect to the strategies described in the previous section.

5.1 Path Construction and Representation

Our main purpose in this study is to generate a common path for users to travel in web pages. In order to define the traveling path, first of all we need to define a data structure for the roles in the path. Since roles in the path are in a sequential order, in which it is possible to navigate backward and forward, array is a convenient structure for our case. In the array, all the roles are included in an order which refers to the order of access to the particular role. However, our initial data are not in the form of heuristic roles, rather the data only includes the names of visual elements in the segmentation results of corresponding web pages. Therefore, a further process of application of role detection on these web pages is required. In other words, we need to construct a single common path of roles, by using common scanpaths of visual elements, which we retrieved from eMINE Scanpath Analysis Algorithm. Experiments of Eraslan et al (2013) showed that, eMINE Scanpath Analysis Algorithm is an efficient and accurate way of detecting the common scanpath. Therefore, as a result, our initial step is to calculate a common scanpath for the roles by using eMINE Scanpath Analysis Algorithm.

However, restricting the users to only travel in constructed path may decrease the accessibility, since some of the roles and their instances may not be included in this path. Therefore, we need to provide an alternative list of visual elements in a web page, which can be considered as a summary of the whole page or a specific visual element. If we propose a single path, it may fail when user jumps to a visual element which is not in the path. One possible solution might be to reset the journey of the user and jump to the initial visual element of the page, however it is not reasonable since user would need to start traveling the web page from the beginning of the path and visit the visual elements he/she already visited. Therefore, probabilistic approaches are more convenient for this purpose, since they enable us to decide with respect to the current status of the user. For this purpose, we need to store each role with its next possible role, rather than just a list of roles in the path. However, the calculation of next visual element may be costly if we will calculate it in each execution. This can be solved by generating a main path and we can also include the set of all the roles and their possible next roles. Analysis of Eraslan et al (2013) indicated that, transition matrix is the most convenient way for this procedure. Transition Matrix is a technique which calculates the probabilities between items to predict a behavior (Markov, 1971). In conclusion, in order to be able to navigate user even if they jump out of the role path, Transition Matrix is a suitable way of producing alternative paths.

At the end of eMINE Scanpath Analysis Algorithm and Transition Matrix implementations, we have both a main path and a number of alternative paths for roles in the knowledge

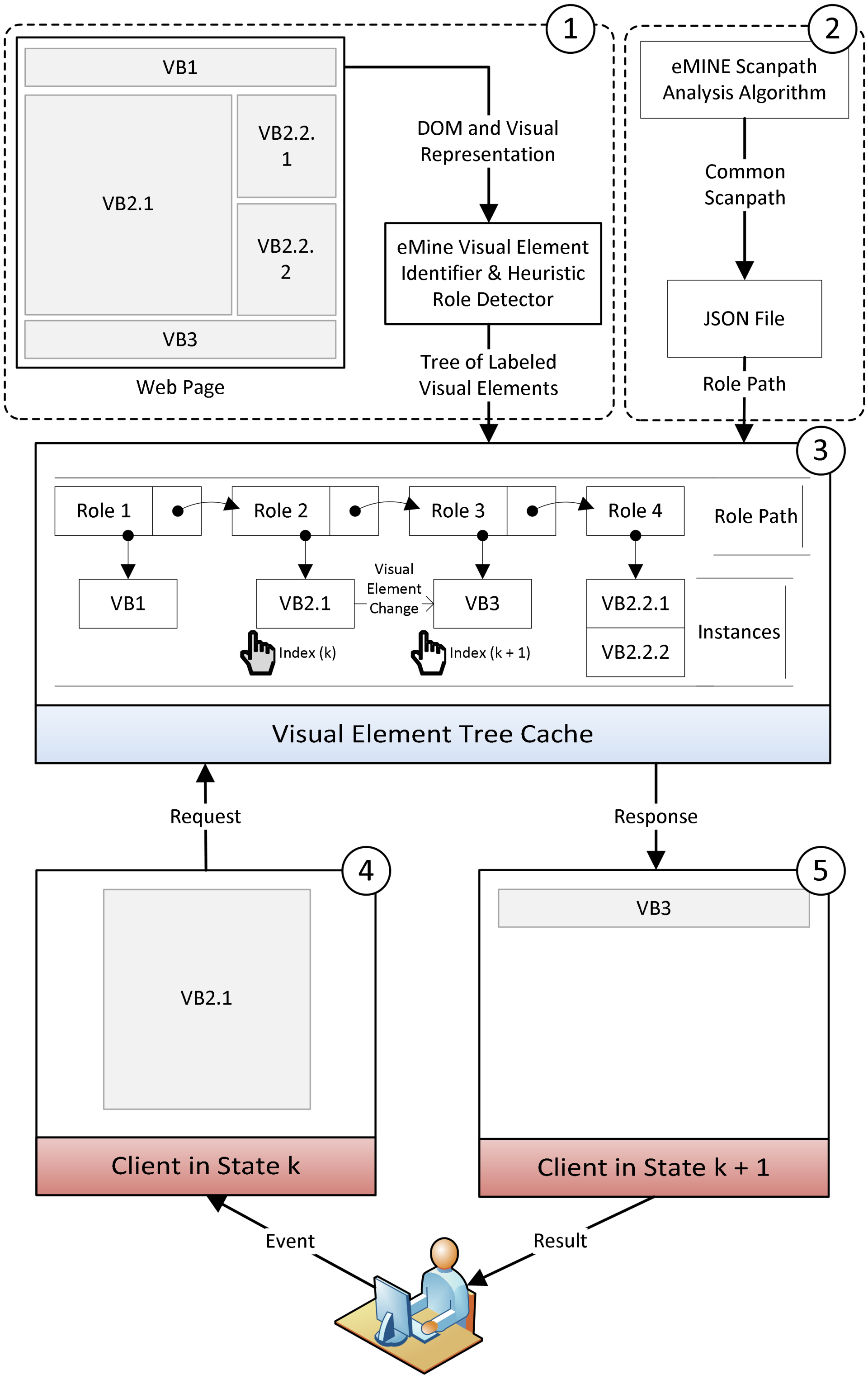


Figure 4: Proxy server application architecture

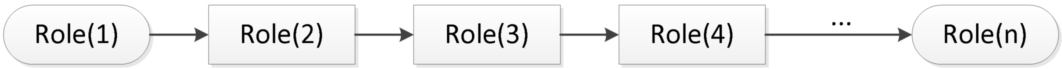


Figure 5: Main role path

base of visual element identification algorithm. We anticipate that, changing web trends and further eye tracking studies may produce a new common role path. Therefore, rather than hard-coding the path in the transcoding implementation, we need a modifiable data storage for the path. Ontology structure is one possible data structure, however, the path definition is not very complex and ontology parsing may result in a costly process. Rather, we can use a simple JSON object in the following structure:

```
{
  "main_path": [
    role-1,
    role-2,
    role-3,
    ...,
    role-n
  ],
  "alternatives": [
    {role-k: role-l},
    {role-x: role-y},
    ...
  ]
}
```

This data structure simply defines a path for user navigation and a set of alternative roles and their next possible roles in case user jumps to a visual element which is not in the main path. Figure 5 illustrates the main path of the travel.

Alternative way of thinking about web pages, is to consider them as a composite structure that includes a set of visual elements. These visual elements can be considered as the instances of the roles in the path. Our experiments with roles of visual elements have shown that, a web page may have multiple instances. Therefore, we need to point a set of visual elements with a role, rather than one-to-one matching. Figure 6 shows the instance set of each role in the path.

5.2 Events

In this study, user interactions for page navigation are referred as events. In desktop, basic events are keyboard inputs and mouse clicks. In mobile devices, basic events are button clicks and screen touches. In this section, we categorize basic event types required to allow user to navigate between visual elements in a web page.

Navigate to the next role: Based on the current status of the user in a web page, user jumps to the next role in the path. If the user is at the final role in the path, next role is selected with respect to the alternative paths.

Navigate to the previous role: Based on the current status of the user in a web page, user jumps to the previous role in the path. If the user is at the initial role in the path, previous role is selected with respect to the alternative paths.

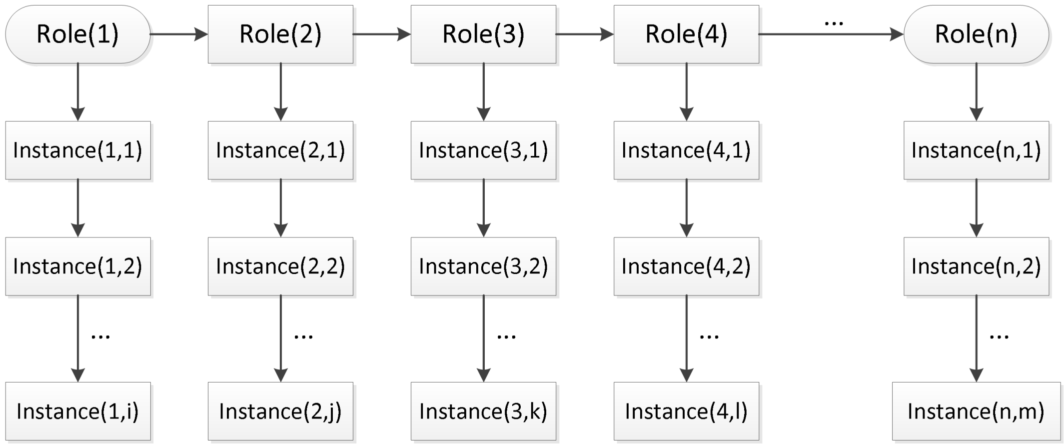


Figure 6: Projection over visual elements

Navigate to the next instance in a specific role: User jumps to the next instance of current role. If the user is at the final instance of the role, then user jumps to the next role.

Navigate to the previous instance in a specific role: User jumps to the previous instance of current role. If the user is at the initial instance of the role, then user jumps to the previous role.

Navigate to the initial role: User jumps to the initial role in the path.

Navigate to the final role: User jumps to the final role in the path.

Display a summary of the page: User is provided a list of each visual element in the page. The visual elements are not listed with respect to their content, rather they are listed with their roles in the page.

Select a visual element in the page: In the summary of the web page, user specifies the index of his/her selection. Then, user jumps to the selected role.

Display a summary of the path: User is provided a list of each visual element in the path. The visual elements are not listed with respect to their content, rather they are listed with their roles in the page.

Select a visual element in the path: In the summary of the path, user specifies the index of his/her selection. Then, user jumps to the selected role.

Display a summary of the instances of a specific role: User is provided a list of instances of the current role.

Select a visual element in a specific role: In the summary of the instances, user specifies the index of his/her selection. Then, user jumps to the selected instance.

Jump to a specific role: User may jump to a specific role, such as menu, immediately.

These events are fired with a set of links provided in the output of each state. Also, some specific event firing methods can be defined in client applications, such as key presses. In order to sum up, we defined a basic user interaction model to navigate in the web page by using both the aspects of eye tracking study and the roles of the visual elements in the web page. These interaction model, then, will be used for selecting the next state visual element in the proxy side architecture.

5.3 Proxy Server Architecture

In Section 5.1, we have defined a data structure for the roles in the common scanpath and in Section 5.2, we have identified the events which will be used for user interaction. In this section, we describe the system architecture for web page transcoding based on eye tracking.

In order to overwhelm the accessibility problems of both screen readers and mobile devices, a proxy based approach is proposed. Such an approach is the most convenient for our application since it enables selective representation, sequential and direct access, as well as, summarization. Figure 7 represents a summary of overall network architecture. In this architecture, mobile users via tablets or smart phones and disabled users via laptops or PCs, send a request in form of either a URL as initial request or an event. Proxy application in the proxy server receives the request. If the request is a URL, it sends a request to the web server to retrieve the overall content of the web page, including style sheets and script files, and process the web page. Also, in this step, proxy application assigns a unique ID to the user and returns it. If the request is in the form of an event, the proxy application selects the next visual element to display with respect to the event and user ID, and returns the visual element to the client application.

In proxy application, the web page is processed to navigate user (see Figure 4). The application takes two inputs. The first input is the role path, which is stored in a JSON file. This path is generated from scanpaths via eMINE Scanpath Analysis Algorithm. The second input is the request from the user. If the request is in the form of a URL, application requests for the web page. When it retrieves the page content, it segments the web page and assigns the role of each visual element in the web page. These visual elements, then, are appended to a list which corresponds to the specific role of the visual element. In initial state, the current index is the first visual element of the first role in the path.

After initial request is processed and a visual element in the web page is returned to the user, user may request a further visual element to represent. In such cases, multiple segmentation and role detection calls for the same page within a session, may cause performance problems. Therefore, it is required to cache the segmentation result of the web page in the proxy server. When proxy application receives a request in the form of an event, it changes the current index with respect to the event type. For example, if the event indicates that, user wants to navigate to the next role, current index is moved to the first instance of the next role. Current visual element is returned to the user after each request. The visual element tree and role instance lists are kept in the proxy server for a specific amount of time, similar to the sessions, and removed when it expires.

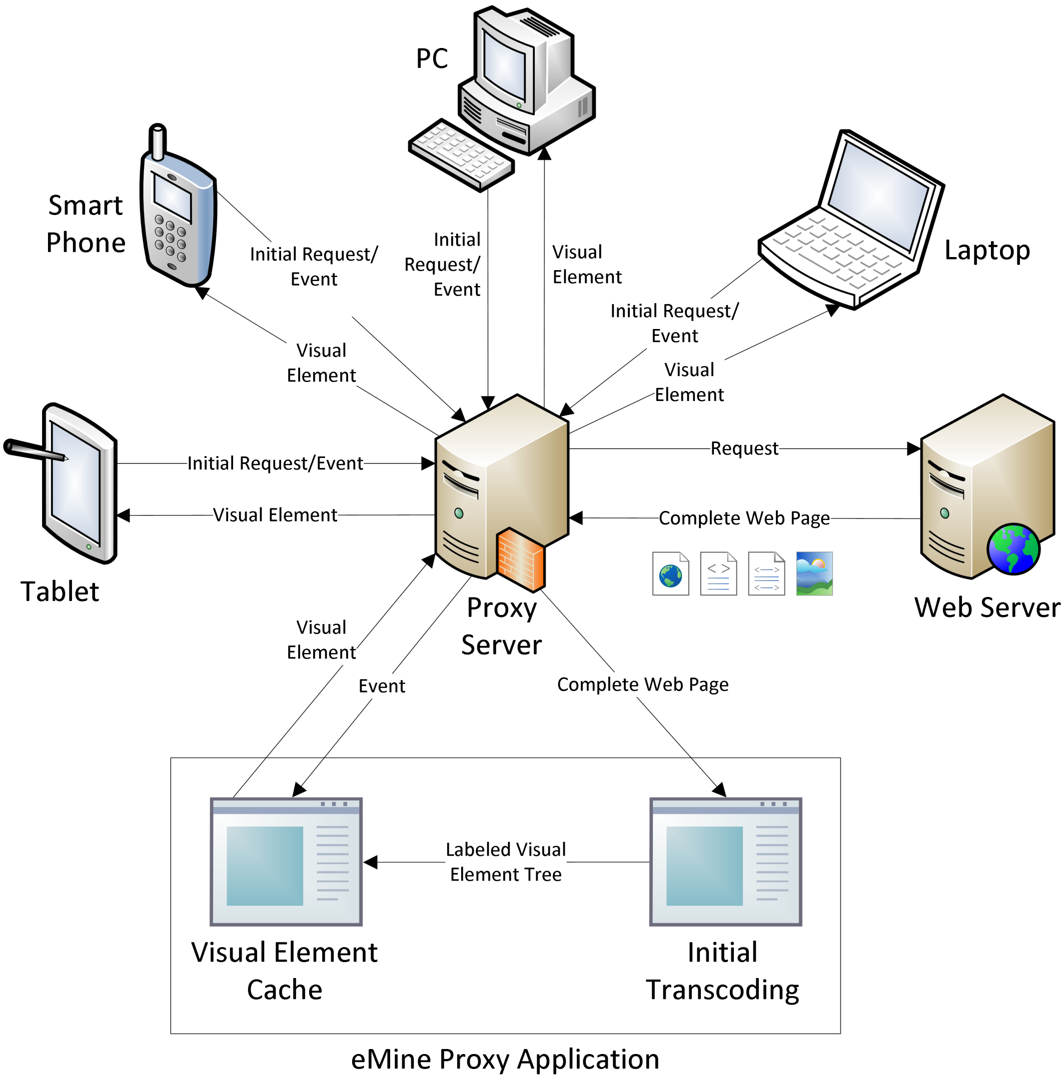


Figure 7: Overall architecture

6 Discussion

We are aware of the fact that, although many web pages have some problems with accessibility and mobile devices, some of them are designed and developer as user friendly for screen readers and mobile devices. In such cases, further transcoding the page may result in performance and accessibility loss. Therefore, a pre-validation of accessibility is important. However, this validation needs a wide range of metrics for both screen readers and mobile devices. In this study, we assume that all pages fail in this validation and we will cover it in our future work.

Our transcoding approach are proposed to be handled in a proxy-based architecture. Advantages of such an approach are independence of content provider and client applications, and dedication of resources for only transcoding task. However, the connection speed between client and proxy, as well as between proxy and server is a major drawback. Alternative approaches may implement the system in client-side or server-side. The major advantages of client approach is that, it enables users to specify their preferences so that, transcoding may fit better in the needs of the users. However, the performance of the client application depends on the resources of the client devices, including processing capacity, as well as bandwidth. Therefore, it is not a suitable approach for mobile device adaptation. Other alternative is server-side implementation, which requires minimum bandwidth usage, since it provides direct access to the source code and requires minimum data transfer. However, this approach can be applied only to privately owned content, which means, it is applicable on a limited number of pages on the web. As we have discussed, proxy, client and server based approaches have both advantages and disadvantages. In order to satisfy our requirements for both mobile device adaptation and web accessibility, proxy based approach is more suitable. With a proxy based approach, we have access to the public content on the web, as well as minimum processing and data transfer in client devices.

7 Conclusion

This technical report presented our eye tracking based transcoding approach. With this approach, we aim to solve both accessibility problems of disabled people and provide efficient browsing in mobile devices. The main contribution of this study is that, it takes both the role of the visual elements in the page layout and user travel behavior, which is presented in a common scanpath as the result of eye tracking studies with web pages.

The architecture of proposed system consists of a knowledge base for common scanpath, a visual element identifier to segment web pages and detect the role of each visual element in the page, a visual element tree cache for storing the visual elements of a web page during the travel of a user, and a software interface to receive user interactions and return the current visual element in the scanpath. User interactions enable user to navigate in both different roles and different instances of a role, as well as jump to specific roles or visual elements.

In conclusion, we proposed an eye tracking based web page transcoding model for both web accessibility and mobile device adaptation. Further studies will also provide a better understanding of common scanpath based on eye tracking data of user experiments. In such cases, our approach is modifiable to adapt different common scanpath outcomes.

References

Adzic V, Kalva H, Furht B (2011) A survey of multimedia content adaptation for mobile devices. Multimedia Tools and Applications 51:379–396

Ahmadi H, Kong J (2008) Efficient web browsing on small screens. In: Proceedings of the working conference on Advanced visual interfaces, ACM, New York, NY, USA, AVI '08, pp 23–30

Ahmadi H, Kong J (2012) User-centric adaptation of web information for small screens. J Vis Lang Comput 23(1):13–28

Akpınar E, Yeşilada Y (2012) Vision based page segmentation: Extended and improved algorithm. Technical report, Middle East Technical University Northern Cyprus Campus

Akpınar E, Yeşilada Y (2013a) Automatic discovery of visual elements of web pages. Technical report, Middle East Technical University Northern Cyprus Campus

Akpınar E, Yeşilada Y (2013b) Heuristic role detection of visual elements of web pages. In: Daniel F, Dolog P, Li Q (eds) Web Engineering, Lecture Notes in Computer Science, vol 7977, Springer Berlin Heidelberg, pp 123–131

Asakawa C, Takagi H (2000) Annotation-based transcoding for nonvisual web access. In: ASSETS'00, ACM Press, pp 172–179

Baluja S (2006) Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework. In: WWW '06: Proceedings of the 15th international conference on World Wide Web, ACM, New York, NY, USA, pp 33–42

Borodin Y, Mahmud J, Ramakrishnan IV, Stent A (2007) The hearsay non-visual web browser. In: Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A), ACM, New York, NY, USA, W4A '07, pp 128–129

Chen J, Zhou B, Shi J, Zhang H, Wu Q (2001) Function-based object towards website adaptation. In: Proceedings of the Tenth International World Wide Web Conference, ACM, Hong Kong

Chen T, Yesilada Y, Harper S (2009) What input errors do you experience? typing and pointing errors of mobile web users. International Journal of Human-Computer Studies, Elsevier 68:138–157

Chen Y, Ma W, Zhang H (2003) Detecting web page structure for adaptive viewing on small form factor devices. In: Proceedings of the Twelfth International World Wide Web Conference

Chen Y, Xie X, Ma WY, Zhang HJ (2005) Adapting web pages for small-screen devices. IEEE Internet Computing 9:50–56

Edwards ADN (2008) Assistive technologies. In: Harper S, Yesilada Y (eds) Web Accessibility: A Foundation for Research, 1st edn, Human-Computer Interaction Series, Springer, London, chap 10, pp 142–162

Eraslan S, Yesilada Y, Harper S (2013) emine scanpath analysis algorithm. Technical report, Middle East Technical University Northern Cyprus Campus and University of Manchester

Harper S, Bechhofer S, Lunn D (2006) Taming the inaccessible web. In: Proceedings of the 24th annual ACM international conference on Design of communication, ACM, New York, NY, USA, SIGDOC '06, pp 64–69

Hattori G, Hoashi K, Matsumoto K, Sugaya F (2007) Robust web page segmentation for mobile terminal using content-distances and page layout information. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, ACM Press, New York, NY, USA, pp 361–370

Hwang Y, Kim J, Seo E (2003) Structure-aware web transcoding for mobile devices. IEEE Internet Computing 7(5):14–21

Lunn D, Bechhofer S, Harper S (2008a) The sadie transcoding platform. In: W4A '08: Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A), ACM, New York, NY, USA, pp 128–129

Lunn D, Bechhofer S, Harper S (2008b) A user evaluation of the sadie transcoder. In: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility, ACM, New York, NY, USA, Assets '08, pp 137–144

Mahmud J, Borodin Y, Das D, Ramakrishnan IV (2007a) Combating information overload in non-visual web access using context. In: Proceedings of the 12th international conference on Intelligent user interfaces, ACM, New York, NY, USA, IUI '07, pp 341–344

Mahmud JU, Borodin Y, Ramakrishnan IV (2007b) Csurf: a context-driven non-visual web-browser. In: Proceedings of the 16th international conference on World Wide Web, ACM, New York, NY, USA, WWW '07, pp 31–40

Markov A (1971) Extension of the Limit Theorems of Probability Theory to a Sum of Variables Connected in a Chain. In: Howard R (ed) Dynamic Probabilistic Systems (Volume I: Markov Models), John Wiley & Sons, Inc., New York City, chap Appendix B, pp 552–577

Milic-Frayling N, Sommerer R (2002) Smartview: Flexible viewing of web page contents. In: Poster Proceedings of the Eleventh International World Wide Web Conference

Song R, Liu H, Wen JR, Ma WY (2004) Learning block importance models for web pages. In: Proceedings of the 13th international conference on World Wide Web, ACM, New York, NY, USA, WWW '04, pp 203–211

Takagi H, Asakawa C, Fukuda K, Maeda J (2002) Site-wide annotation: Reconstructing existing pages to be accessible. In: ASSETS'02, ACM Press, pp 81–88

Whang Y, Jung C, Kim J, Chung S (2001) Webalchemist: A web transcoding system for mobile web access in handheld devices. In: Optoelectronic and Wireless Data Management, Processing, Storage, and Retrieval, pp 102–109

Xiang P, Yang X, Shi Y (2007) Web page segmentation based on gestalt theory. In: Multimedia and Expo 2007 IEEE International Conference (ICME)

Xiao X, Luo Q, Hong D, Fu H (2005) Slicing*-tree based web page transformation for small displays. In: Proceedings of the 14th ACM international conference on Information and knowledge management, ACM, New York, NY, USA, CIKM '05, pp 303–304

Xiao Y, Tao Y, Li Q (2008a) Web page adaptation for mobile device. In: Wireless Communications, Networking and Mobile Computing

Xiao Y, Tao Y, Li W (2008b) A dynamic web page adaptation for mobile device based on web2.0. In: Proceedings of the 2008 Advanced Software Engineering and Its Applications, IEEE Computer Society, Washington, DC, USA, pp 119–122

Xie X, Miao G, Song R, Wen JR, Ma WY (2005) Efficient browsing of web search results on mobile devices based on block importance model. In: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, IEEE Computer Society, Washington, DC, USA, pp 17–26

Yang X, Shi Y (2009) Enhanced gestalt theory guided web page segmentation for mobile browsing. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03, IEEE Computer Society, Washington, DC, USA, WI-IAT '09, pp 46–49

Yesilada Y, Stevens R, Harper S, Goble C (2007) Evaluating DANTE: Semantic transcoding for visually disabled users. ACM Trans Comput-Hum Interact 14(3):14

Yesilada Y, Harper S, Eraslan S (2013) Experiential transcoding: an eyetracking approach. In: Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility, ACM, New York, NY, USA, W4A '13, pp 30:1–30:4

Yin X, Lee W (2004) Using link analysis to improve layout on mobile devices. In: Proceedings of the Thirteenth International World Wide Web Conference, pp 338–344

Yin X, Lee WS (2005) Understanding the function of web elements for mobile content delivery using random walk models. In: Special interest tracks and posters of the 14th international conference on World Wide Web, ACM, New York, NY, USA, WWW '05, pp 1150–1151

Zakas NC (2013) The evolution of web development for mobile devices. Commun ACM 56(4):42–48