

AUTOMATIC IDENTIFICATION AND ROLE DETECTION OF VISUAL
ELEMENTS IN WEB PAGES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET ELGIN AKPINAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY 2014

Approval of the thesis:

**AUTOMATIC IDENTIFICATION AND ROLE DETECTION OF VISUAL
ELEMENTS IN WEB PAGES**

submitted by **MEHMET ELGIN AKPINAR** in partial fulfillment of the requirements
for the degree of **Master of Science in Computer Engineering Department, Middle
East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Pınar Karagöz
Supervisor, **Computer Engineering Department, METU**

Assist. Prof. Dr. Yeliz Yeşilada
Co-supervisor, **Computer Engineering Dept., METU NCC**

Examining Committee Members:

Prof. Dr. Faruk Polat
Computer Engineering Department, METU

Assoc. Prof. Dr. Pınar Karagöz
Computer Engineering Department, METU

Assist. Prof. Dr. Yeliz Yeşilada
Computer Engineering Department, METU

Dr. Ayşenur Birtürk
Computer Engineering Department, METU

Dr. Alev Mutlu
Computer Engineering Department, Selçuk University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MEHMET ELGIN AKPINAR

Signature :

ABSTRACT

AUTOMATIC IDENTIFICATION AND ROLE DETECTION OF VISUAL ELEMENTS IN WEB PAGES

Akpınar, Mehmet Elgin

M.S., Department of Computer Engineering

Supervisor : Assoc. Prof. Dr. Pınar Karagöz

Co-Supervisor : Assist. Prof. Dr. Yeliz Yeşilada

July 2014, 164 pages

Web pages typically include many visual elements to support interaction with the user. However, if visual elements do not comply with web accessibility guidelines, they become inaccessible in alternative presentations, such as audio. This study presents an automatic role detection approach to identify visual elements in web pages and their roles. The system architecture has three major components: automatic identification of visual elements in web pages; automatic generation of heuristic rules from the knowledge base; and application of these rules for automatic annotation of visual elements with their roles. Moreover, automatic role detection approach is deployed in a case study of web page transcoding based on eye tracking. This thesis first explains the system architecture in detail and then presents both technical and user evaluations of the proposed approaches. User evaluation shows that the automatic role detection approach has around 80% receptive accuracy, but the proposed knowledge base could be further improved for better results. User evaluation also shows that the transcoding application partially improves the information quality of web pages.

Keywords: Web Accessibility, Mobile Web, Knowledge Engineering, Ontology, Heuristics, Rule Based Systems, Transcoding

ÖZ

WEB SAYFALARINDAKİ GÖRSEL ELEMANLARIN OTOMATİK BELİRLENMESİ VE ROLLERİNİN TESPİT EDİLMESİ

Akpınar, Mehmet Elgin

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi : Doç. Dr. Pınar Karagöz

Ortak Tez Yöneticisi : Yrd. Doç. Dr. Yeliz Yeşilada

Temmuz 2014 , 164 sayfa

Web sayfaları genel olarak bir çok görsel eleman içermektedir. Bu görsel elemanlar web erişilebilirlik standartlarının uygulanmadığı durumlarda ekran okuyucu gibi alternatif platformlarda erişilemez hale gelmektedir. Bu çalışmada web sayfalarında yer alan görsel elemanları ve bu görsel elemanların rollerini belirleyen bir yaklaşım öne sürülmüştür. Sistem mimarisi üç ana bileşenden oluşmaktadır: sayfa içindeki görsel elemanların otomatik belirlenmesi, bilgi bankasından kuralların otomatik oluşturulması ve bu kuralların görsel elemanların rollerini belirlemek amacıyla uygulanması. Ayrıca öne sürülen otomatik rol belirleme yaklaşımı göz izleme tabanlı web sayfası dönüştürme işleminde kullanılmıştır. Bu çalışma öncelikle öne sürülen sistem mimarilerini detaylı şekilde açıklamakta; daha sonra teknik değerlendirme ve kullanıcı deneyleri süreçlerini anlatmaktadır. Yapılan deneyler sonucunda, otomatik role belirleme sisteminin %80 doğrulukta sonuçlar verdiği görülmüştür. Fakat bu sonuçlar, bilgi bankasının geliştirilmesiyle iyileştirilebilmektedir. Ayrıca kullanıcı değerlendirmeleri yapılan dönüştürme işleminin web sayfalarının bilgi kalitesini arttırdığı görülmüştür.

Anahtar Kelimeler: Web Erişilebilirliği, Mobil Web, Bilgi Mühendisliği, Ontoloji, Bulgusal Kurallar, Kural Tabanlı Sistemler, Dönüştürme

To my family

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Yeliz Yeşilada for all her continuous support, her patience and motivation. Her guidance and positive behavior helped me in all the time of research and writing of this thesis.

Besides my advisors, I would like to thank the examining committee members: Pınar Karagöz, Faruk Polat, Ayşenur Birtürk and Alev Mutlu for their precious time reading my thesis and for their constructive comments.

I would like to thank to my employer in Nart Informatics, Taha Yayıcı, for providing me with time to study whenever I needed it.

My special thanks go to Şükrü Eraslan, who helped me to improve this study with his invaluable feedbacks and bug reports. I also wish to thank Aslı Niyazi and Cansu Telkes for their assistance with the statistics used in this thesis.

I would also like to thank my family: Zübeyde Akpınar, Hasan Akpınar and Ekin Akpınar for their unconditional love and support. My deepest thanks are to Nihan Özdemirler, who motivated me with her never ending patience, tolerance and understanding throughout this study.

Finally, this study was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) with the grant number 109E251. As such I would like to thank to (TÜBİTAK) for their financial support.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vi
ACKNOWLEDGMENTS	viii
TABLE OF CONTENTS	ix
LIST OF TABLES	xvi
LIST OF FIGURES	xix
LIST OF ABBREVIATIONS	xxi
CHAPTERS	
1 INTRODUCTION	1
1.1 Contributions of Thesis	4
1.2 Thesis Layout	4
2 BACKGROUND WORK	7
2.1 Web Basics	7
2.2 Web Accessibility	9
2.3 Segmentation	11
2.4 Transcoding	13

2.5	Ontologies	15
2.6	Rule Based Systems	19
2.7	Eye Tracking	21
2.8	Summary	22
3	RELATED WORK	25
3.1	Web Page Segmentation	25
3.1.1	HTML Tag Based Approaches	26
3.1.2	Position and Alignment Based Approaches	26
3.1.3	Size Based Approaches	27
3.1.4	Vision Based Approaches	28
3.1.5	Machine Learning Approaches	29
3.1.6	Repetition Based Approaches	29
3.1.7	Custom Approaches	29
3.2	Heuristic Role Detection of Visual Elements	30
3.3	Web Page Transcoding	32
3.3.1	Web Accessibility	32
3.3.2	Small Screen Adaptation	33
3.4	Summary	34
4	AUTOMATIC IDENTIFICATION OF VISUAL ELEMENTS IN WEB PAGES	37
4.1	System Architecture Overview	37
4.2	Visual Element Identification	39

4.2.1	VIPS Algorithm and Its Limitations	41
4.2.2	VIPS Improvements	45
4.3	Automatic Role Detection	48
4.3.1	Rule Generation	49
4.3.2	Role Detection	58
4.4	Implementation	60
4.5	Summary	61
5	WEB PAGE TRANSCODING	63
5.1	Transcoding Strategies	63
5.1.1	Selective Representation	64
5.1.2	Summarization	64
5.1.3	Sequential and Direct Access	65
5.2	Eye Tracking Based Transcoding	66
5.3	System Architecture	70
5.4	Implementation of Transcoding Application	71
5.4.1	Block Based Transcoding	73
5.4.2	Role Based Transcoding	74
5.5	Summary	77
6	EVALUATION	81
6.1	User Evaluation of Automatic Role Detection	81
6.1.1	Materials	82

6.1.2	Administration	83
6.1.3	Procedure	83
6.1.4	The Scientific Questions	84
6.1.5	Participants	85
6.1.6	Results	86
6.1.6.1	Techniques of Evaluation	86
6.1.6.2	Rating Results for Visual Element Identification	87
6.1.6.3	Ranking Results for Visual Element Identification	89
6.1.6.4	Correlation Between Rating and Ranking Results	90
6.1.6.5	Evaluation Results of Automatic Role Detection	91
6.1.7	Discussion	92
6.2	Technical Evaluation of Automatic Role Detection	94
6.2.1	Discussion	95
6.3	User Evaluation of Web Page Transcoding Based on Eye Tracking	96
6.3.1	Materials	96
6.3.2	Administration	97
6.3.3	Procedure	97
6.3.4	Hypothesis	99
6.3.5	Participants	100

6.3.6	Results	101
6.3.6.1	Techniques of Evaluation	101
6.3.6.2	Overall Test Results	102
6.3.6.3	Task Completion, System Usability, Information and Interface Quality Test Results	103
6.3.6.4	Complexity Based Test Results and Com- parison	104
6.3.6.5	Correlation Between Test Results and Complexity	105
6.3.6.6	Demographics Based Test Results and Comparison	106
6.3.6.7	Correlation Between Test Results and Demographics	107
6.3.6.8	Correlation Between Tests	107
6.3.7	Discussion	108
6.4	Technical Evaluation of Web Page Transcoding Based on Eye Tracking	111
6.4.1	Access Time Comparison Results	111
6.4.2	Visual Comparison Results	112
6.4.3	Mobile Simulation Results	113
6.4.4	Discussion	114
7	CONCLUSION AND FUTURE WORK	117
7.1	Future Work	119
	REFERENCES	121

APPENDICES

A	HTML AND HTML5 TAG CLASSIFICATION	133
A.1	HTML Tag Classification	133
A.2	HTML5 Tag Classification	133
B	MODIFIED RULE SET FOR VISUAL ELEMENT IDENTIFICATION	135
C	CLASS DEFINITIONS FOR ROLE PROPERTIES	137
D	OBJECT PROPERTIES	139
E	COMMON SCANPATHS	141
E.1	www.apple.com	141
E.2	www.avg.com	142
E.3	uk.yahoo.com	143
E.4	www.godaddy.com	144
E.5	www.bbc.co.uk	145
F	EVALUATION RESULTS OF VISUAL ELEMENT IDENTIFICATION	147
F.1	Rating Results With Respect to User Profiles	147
F.2	Correlation Between Rating Results and Participant Profiles .	150
F.3	Ranking Results With Respect to User Profiles	151
F.4	Correlation Between Ranking Results and Participant Profiles	156
G	DEMOGRAPHIC INFORMATION QUESTIONNAIRE	157
H	USER EXPERIMENT QUESTIONNAIRE ITEMS	159
I	EVALUATION RESULTS OF WEB PAGE TRANSCODING	161

I.1	Item Based Test Results of Original and Transcoded Pages . .	161
I.2	Transcoding Evaluation Results With Respect to User Profiles	162
I.3	Correlation Between Transcoding Evaluation Results and Participant Profiles	164

LIST OF TABLES

TABLES

Table 5.1	Role paths of six web pages	69
Table 6.1	Web pages used in the user study and their complexity scores	82
Table 6.2	Demographic summary of participants	85
Table 6.3	Overall rating results	87
Table 6.4	Rating results for complexity groups with 5 levels	87
Table 6.5	Paired T Test results on rating data	88
Table 6.6	Ranking results for pages with 3 and 4 levels	89
Table 6.7	Ranking results for pages with 5 and more levels	89
Table 6.8	Ranking results for complexity groups	90
Table 6.9	Pairwise T Test results on ranking data	91
Table 6.10	Rating * ranking crosstabulation	92
Table 6.11	Chi-Square Test results	92
Table 6.12	Accuracy results of role detection algorithm	92
Table 6.13	Performance results for visual element identification	94
Table 6.14	Performance results for automatic role detection	95
Table 6.15	Searching tasks used in the study	98
Table 6.16	Demographic summary of participants	100
Table 6.17	Test results for original and transcoded pages	104
Table 6.18	Test results for visual complexity groups	105
Table 6.19	Test results for visual complexity groups	105

Table 6.20 Correlation with visual complexity	106
Table 6.21 Correlation with demographics	107
Table 6.22 Correlation between tests	108
Table 6.23 Access times to visual elements in common scanpaths	112
Table 6.24 Visual comparison test results	113
Table 6.25 Mobile simulation tests results	114
Table A.1 HTML tag classification	133
Table A.2 HTML5 tag classification	133
Table F.1 Rating results for pages with 3 levels with respect to user profiles . .	147
Table F.2 Rating results for pages with 4 levels with respect to user profiles . .	148
Table F.3 Rating results for pages with 5 and more levels with respect to user profiles	149
Table F.4 Pearson's Chi Square Test results for participant groups	150
Table F.5 Ranking results for pages with 3 levels, with respect to user profiles	151
Table F.6 Ranking results for pages with 3 levels, with respect to user profiles (cont.)	152
Table F.7 Ranking results for pages with 4 levels, with respect to user profiles (Part 1: Age and gender)	152
Table F.8 Ranking results for pages with 4 levels, with respect to user profiles (Part 2: Education, level of expertise and current status in web design and development)	153
Table F.9 Ranking results for pages with 5 and more levels, with respect to user profiles (Part 1: Age, gender and level of expertise)	154
Table F.10 Ranking results for pages with 5 and more levels, with respect to user profiles (Part 2: Education and current status in web design and de- velopment)	155
Table F.11 Pearson's Chi Square Test results for participant groups	156
Table I.1 Item based test results of original and transcoded pages	161

Table I.2	Test results for demographic groups	162
Table I.3	Test results for demographic groups	163
Table I.4	Test results for participant groups (N: Normal, NN: Not Normal) . .	164

LIST OF FIGURES

FIGURES

Figure 2.1	Sample segmentation of timeanddate.org page	12
Figure 4.1	System Architecture	38
Figure 4.2	Comparison of DOM based and vision based segmentation	40
Figure 4.3	Visual elements and tree structure for Wordpress.org	43
Figure 4.4	A typical layout for web pages	47
Figure 4.5	Two identical title elements with different HTML tags	49
Figure 4.6	Visual elements in ACTF Home Page	53
Figure 4.7	Overall results for ACTF Home Page	60
Figure 5.1	eMINE - Home page in both desktop and mobile	65
Figure 5.2	Babylon Home Page	68
Figure 5.3	Projection over visual elements	70
Figure 5.4	Proxy server application architecture	72
Figure 5.5	eMINE - Home page in block based transcoding application	73
Figure 5.6	Role Based Transcoding Demo Application	75
Figure 5.7	eMINE - Home page in role based transcoding demo application	77
Figure 5.8	eMINE - Home page in role based transcoding demo application	78
Figure 5.9	Direct access to menu	79
Figure 6.1	Original and transcoded pages used in the study	97
Figure E.1	Apple Home Page	141

Figure E.2	AVG Home Page	142
Figure E.3	Yahoo Home Page	143
Figure E.4	Godaddy Home Page	144
Figure E.5	BBC Home Page	145

LIST OF ABBREVIATIONS

ACTF	Accessibility Tools Framework
API	Application Programming Interfaces
ARIA	Accessible Rich Internet Applications
ASQ	After-Scenario Questionnaire
CSS	Cascading Style Sheet
DOM	Document Object Model
HTML	Hypertext Markup Language
INFOQUAL	Information Quality Test
INTERQUAL	Interface Quality Test
JSON	JavaScript Object Notation
OWL	Web Ontology Language
PSSUQ	Post-Study System Usability Questionnaire
RDF	Resource Description Framework
RDFS	RDF Schema
SYSUSE	System Usefulness Test
VCS	Visual Complexity Score
VE	Visual Element
VICRAM	Visual Complexity Rankings and Accessibility Metrics
VIPS	Vision Based Page Segmentation
W3C	World Wide Web Consortium
WAfA	Web Authoring for Accessibility
XML	Extensible Markup Language
XPATH	XML Path Language

CHAPTER 1

INTRODUCTION

The World Wide Web (web) has been a major source of information and services in people's daily lives. Web pages consist of many visual elements in order to guide the readers. Typical design process aims to provide visually interactive elements in web pages. As web technologies evolve, they come up with more useful and powerful tools and libraries to provide better user experience. Using these tools, designers and developers are able to create more visually interactive web pages. Consequently, modern web pages have technically more sophisticated layouts and structures. However, when it comes to web accessibility, this evaluation of web introduced many significant issues. In case of inattention to web accessibility guidelines, these interactive visual elements become unavailable when pages are accessed in alternative forms, such as in audio with assistive technologies. As a result, web pages become inaccessible.

According to World Health Organization (WHO), there are 285 million visually disabled people worldwide [94]. To address web accessibility issues of visually disabled people, web community has made various efforts, including publication of web accessibility guidelines. However, practices to create accessible web pages are optional; therefore, most designers do not follow these guidelines. Web accessibility guidelines are very important, since they conduct how web pages and visual elements are processed by assistive technologies; such as main headlines or alternative presentation of graphical content. According to Disability Rights Commission (DRC), 81% of web pages do not satisfy even basic requirements of web accessibility [57]. Investigation of web accessibility of about 30 million web pages showed that, only around 4% of

the visual elements in these pages complied with the requirements [81]. Moreover, another study which investigated the accessibility of most popular web pages revealed that, only 10% web pages satisfied the web accessibility standards. Web accessibility has also been a social issue. Many governments have made serious efforts to provide accessible services to their citizens. However, there are still major web accessibility issues with some public services. For example, investigation of web accessibility of Central Physician Appointment System page of Turkey showed that, this public health service had very critical web accessibility issues [93]. Similarly, a recent study showed that, web pages of public institutions in Turkey also had many web accessibility problems [39]. Finally, evaluation of web pages of 10 randomly selected Turkish universities showed that, all of these web pages failed to satisfy accessibility requirements completely [74].

Web accessibility is also a major issue for people without disabilities who access web by using alternative platforms, such as mobile devices [57]. One major problem is small screen size of the devices. Although technological innovations set forward the enlargement of screen sizes, they still stay smaller when they are compared to larger screens of desktop PCs or laptops. Consequently, navigating in a web page, which originally designed only for large screens, is an important problem [91, 4]. Another drawback of such devices is their limited memory and bandwidth. Processing visually complex web pages is a costly task [124, 132]. Moreover, modern mobile devices typically do not provide a keyboard or mouse, rather they are interacted with gestures. Therefore, some aspects of keyboard and mouse interaction in desktop are not available in such devices [124, 120, 31]. To overwhelm these limitations, some content providers also produce a mobile mirror of the original design. However, a recent study which investigated the rate of web pages having a mobile mirror showed that, around 68% of the most popular web pages had mobile mirrors; while this rate decreases to about 50% for a sample of randomly selected web pages [103]. In conclusion, these drawbacks of mobile web access arise the requirement of an alternative representation for mobile devices [2].

To solve these limitations and improve the web accessibility, many studies have been conducted in the literature. These studies aim to present web pages in alternative forms. A typical approach in these studies is to examine the visual elements in web

pages along with their corresponding roles to transform the structure of web pages to appropriate forms for screen readers or mobile devices. However, our analysis showed that, there are limitations with these studies. Identifying roles of visual elements enables these roles to be used in applications that aim to improve web accessibility; therefore, having a deep understanding of the structure of web pages along with the roles of visual elements is very important. Unfortunately, there is no work that provides a deep understanding of visual elements and covers a wide range of design patterns.

Role identification requires reverse engineering techniques which are applied on underlying source code of web pages. However, current web technologies have very flexible syntax and it is possible to create the same visual representation with different coding styles. This means that, automating the role identification is a very challenging task. In order to describe each role with its exact characteristics, it requires too many rules to be generated. Although there exist studies in the literature that propose a range of roles for visual elements and some attributes which are used in identification, these approaches tend to be simplistic and do not cover a comprehensive understanding of visual elements.

In this study, we describe a novel approach for automatically detecting roles of visual elements. Our approach consists of a comprehensive role set and knowledge base to describe the characteristics of visual elements. Moreover, it has two major components. First of all, visual elements in web pages are automatically identified by using both the visual rendering and the underlying source code. In this process, we extended and implemented an existing web page segmentation algorithm, which is called Vision Based Page Segmentation (VIPS) Algorithm [25]. After identifying visual elements in web page, the second component detects the corresponding roles of visual elements in a probabilistic approach. The reason for not applying direct reasoning with the knowledge base and the visual elements is due to the flexible nature of web technologies and inability of fully and absolutely describing visual elements and their definite properties. In our approach, a visual element may accomplish all of the requirements of a heuristic role, or it may satisfy a set of properties of the role. The more attributes of a role a visual element satisfies, the more likely that the visual element has the corresponding role in the page. Furthermore, we present a transcod-

ing application based on eye tracking as a case study. This study also presents our technical and user evaluations of the proposed approaches and architectures. Technical evaluations mainly investigated the technical feasibility of proposed automatic role detection and transcoding systems. Moreover, user evaluations aimed to determine perceived success of our proposed approaches. The transcoding application and evaluation results showed that, our visual element identification implementation and automatic role detection approach can be successfully used in a transcoding application to improve web accessibility.

1.1 Contributions of Thesis

This thesis contributes to the literature in several points. First of all, this study gives detailed explanation of extension and implementation of an existing web page segmentation algorithm. During the study, the source code of the implementation was also released as an open source project. A more important contribution of this thesis is the automatic role detection approach. This approach is modifiable for different research purposes and maintainable to new design trends and web technologies. Moreover, it consists of a comprehensive role set and knowledge base to describe the characteristics of visual elements. Another contribution of this study is the transcoding application based on eye tracking. This application demonstrated deployment of both our automatic role detection approach and aspects of eye tracking studies in web page transcoding. Finally, this study presents both technical and user evaluation for automatic role detection and web page transcoding. With this evaluation, the study has gone some way towards enhancing our understanding of perceived success of our approaches, as well as segmentation level preference of users.

1.2 Thesis Layout

The rest of the thesis has been organized as follows:

Chapter 2: Background Work This chapter describes the underlying technologies of web pages, web accessibility, processes to manipulate the structure of web

pages and some concepts which can be used to describe semantic structure of a web page.

Chapter 3: Related Work There are various approaches in the literature to segment web pages, identify visual elements in these pages and transcode them to alternative representations. In this chapter, we review these approaches in detail. Our analysis showed that, although each approach takes different aspects of web pages, they have some limitations.

Chapter 4: Automatic Identification of Visual Elements in Web Pages This chapter presents our novel automatic role detection approach and system architecture in detail. Our approach mainly consists of identification of visual elements, knowledge description and formalization for roles and their characteristics, and automatic detection of the roles of visual elements based on constructed knowledge.

Chapter 5: Web Page Transcoding In this chapter, we present a case study of web page transcoding based on eye tracking to demonstrate a use case for our visual element identification implementation and automatic role detection approach. Our transcoding mechanism differs from the existing techniques since it is based on eye tracking data to better adapt user navigation behaviour on web pages.

Chapter 6: Evaluation In order to validate our automatic role detection and web page transcoding approaches, we conducted both technical and user evaluations. This chapter presents the methods, materials and the results of these evaluations. The results of evaluation process showed that, success of middle and higher levels in visual element identification were perceived as above average. Moreover, participants preferred higher level of granularity in visual element identification. The automatic role detection approach has around 80% receptive accuracy. Finally, the transcoding application partially improves the information quality of web pages.

Chapter 7: Conclusion and Future Work Finally, in this chapter, we conclude the thesis and discuss our future work.

CHAPTER 2

BACKGROUND WORK

To automatically discover semantic structure of web pages and fully describe the elements within that structure, it is essential to understand web technologies and concepts behind it. This chapter begins by describing fundamental web technologies, main aspects of web accessibility, its importance and the standards published to construct accessible pages. Then, we introduce segmentation and transcoding, which are two main processes that we deploy in our system. Finally, basic formal knowledge and rule representation concepts are covered. In particular, basic properties of ontologies and the evaluation of ontology languages, as well as some examples of an ontology language are given. Furthermore, rule based systems, their main components and rule engines are explained. Finally, eye tracking is introduced briefly.

2.1 Web Basics

In 1989, World Wide Web (web) was invented by Tim Berners-Lee at the European Organization for Nuclear Research (CERN) Laboratory, Switzerland, for the purpose of sharing and managing networked information over the Internet [12, 100, 106]. Web architecture consists of two tiers: web servers which create and deliver the information and clients which retrieve and display it [100].

The main constituent of web pages is markup languages, which defines the structure and presentation of a web page. Hypertext Markup Language (HTML) is the original markup language of the web. HTML enables to author the basic structure of a web document, representation of text and media elements and include links between

web pages with hyperlinks. HTML pages consist of HTML elements, which are represented with tags, attributes and content in hierarchical structure and surrounded with angular brackets. HTML documents are text files which contain HTML elements [100]. One major advantage of HTML is that, it is device independent [1]. Although modern web pages are capable of representing a complex visual layout, first web pages had very simple document structure [106].

Berners-Lee proposed the first public version of HTML, which consists of only 20 tags to create simple structured web pages. In 1995, Internet Engineering Task Force introduced HTML 2.0 with new extensions such as form elements [100]. In order to prevent HTML standards to differentiate between different organizations, World Wide Web Consortium (W3C) was founded in 1994 by Berners-Lee. The main intention of W3C was to set standards and recommendations for HTML [35]. In 1997, HTML 3.2 and HTML 4.0 were published by W3C. In 1999, W3C enhanced HTML with HTML 4.01 [100, 106].

In 1997, W3C released Extensible Markup Language (XML), which has stricter rules with respect to HTML. Unlike flexible structure of HTML, XML documents are required to be syntactically well-formed; otherwise, software which renders document cannot process it. W3C inspired by this feature of XML and redesigned HTML in XML based form with all the elements in the final version and proposed Extensible Hypertext Markup Language, or XHTML 1.0. After first draft of XHTML 2.0 which was not backward compatible with previous versions and abolished some common tags, web developer community became unsatisfied with the strictness of XHTML2.0 since existing content on the web would be invalid. Hence, in 2004, an independent community named Web Hypertext Application Technology Working Group (WHATWG) proposed a new version of HTML. In 2007, W3C declared that they would proceed with HTML support and XHTML 2.0 working group was closed down in 2009 [106].

The final version of the HTML, HTML5, aims to address limitations of the previous versions and requirements of modern web applications [106]. It expands the set of tags, such as nav, header, footer. These new tags specify new behaviours and representations, as well as semantics to the HTML elements [96].

The visual representation of a web page is determined by style sheets. These style sheets are documents that specify visual styles of web pages, such as font properties, color, text alignment, spacing and position attributes. Cascading Style Sheet (CSS) is the style sheet language recommended by W3C. CSS is very useful since it enables to separate the definition of visual styles from the document structure and the same styles can be applied to multiple documents. There are three levels of CSS: inline, document level and external. Inline styles are defined within the tag of an element and applies only to corresponding element. Document level styles are defined in the head section of the document and scope of these styles is limited with the corresponding document. Finally, external styles are included in the document in an external file, and the same styles can be applied to different documents. CSS also provide a selector mechanism to be applied only to a set of elements with specific class and id values [100, 104].

The Document Object Model (DOM), is a W3C specification and a language and platform independent APIs in which interfaces are provided for dynamic access and manipulation of content, structure and style in structured XML and HTML documents by programs and scripts. DOM has a tree structure, defining hierarchical relations between nodes, which refer to elements in the document structure [61, 87].

W3C developed XML Path Language (XPath) to access particular parts of XML documents. XPath aims to provide a common mechanism for all applications which identify specific parts of XML documents by defining these parts with a location path. This location path begins with a root node, and includes all antecedent element nodes in the hierarchical tree structure of the document [116].

2.2 Web Accessibility

The term web accessibility refers to that disabled people can use, comprehend, navigate and interact with the web [109]. In general term, providing accessibility is to remove the obstacles which prohibit disabled people from involving in major activities. These activities involve use of particular services, products and information [11]. From web perspective, providing web accessibility helps disabled people access web

content without assistance, which may involve online social services as well as other public information which is accessible by any other people. The term disability comprise not only visual impairments [10], but also cognitive and learning impairments [75], hearing impairments [28], physical impairments [111] and ageing [73]. Web accessibility mainly focuses on all of these aspects of disability.

Web accessibility has been a very important social issue. Governments also sees its significance and make serious efforts to ensure their citizens access web content. For example, in US, the amendment of Section 508, which focuses on accessibility, has been effective since June 2001 [108]. Similarly, in Turkey, Family and Social Policies Ministry also defines accessibility as buildings, public areas and transportation and informative services, communication technologies being securely and independently accessible and usable by disabled people; and enacts laws to ensure it [44].

As web has been a fundamental source of information for many people, inaccessibility which arises from common usage of visual web content has become a very serious disadvantage for disabled people. In order to help visually disabled people access web content, two different approach in assistive technologies have been developed. First approach is to use a specialized browser, which renders the web content with respect to the page structure and underlying source code. The other approach is to use a standard browser with assistive technologies, such as screen reader. Although specialized browsers can analyze the structure of the page and emphasise different structural components, they are no longer developed and supported. There are many screen reader tools. However, they have the disadvantage of not being able to parse the web page. Rather, they only read what the standard browser visually represents [41]. Assistive technologies are very important since they are major tools for disabled people to access web. They mainly benefit some design patterns in the underlying source code of web pages. For example, disabled users are navigated by title elements. Moreover, alternative content, such as an image, is presented by using its alternative text. However, when web pages are not designed and coded properly, assistive technologies cannot identify these patterns and corresponding visual elements become inaccessible.

Web accessibility has been a major concern in W3C. W3C Web Accessibility Initiative (WAI), founded in April 1997 and aims to make web accessible to disabled people

by publishing guidelines and providing resources [62]. WAI approaches accessibility with respect to three basic components of the web: web content, authoring tools and user agents [54]. Web Content Accessibility Guidelines (WCAG) specifies how markup languages and native information such as text and graphical content should be used to provide content which satisfies accessibility requirements [63, 54]. Moreover, Authoring Tool Accessibility Guidelines (ATAG) focus on the software which is used to create web content in two different perspective. First of all, ATAG aims to help disabled people to create web content by using accessible authoring tools. Furthermore, ATAG aims to improve the authoring tools, so that authors can create accessible web content, which satisfies the WCAG [64]. Therefore, ATAG addresses authoring tool developers. Finally, User Agent Accessibility Guidelines (UAAG) aims to make user agents, such as web browsers and assistive technologies accessible [65].

Another aim of W3C is to provide accessible web to people without disabilities on alternative platforms, such as mobile devices. In order to achieve this, W3C publishes a set of standards for mobile devices. These standards include additional features, including CSS media queries which enable to specify different visual representation styles for screens in different size and resolution. However, Web and Mobile Interest Group of W3C still works on these standards and some of them are not stable [37].

Modern web pages include many widgets and behaviours which benefit dynamic web content and technologies such as JavaScript and AJAX. Using such features in accessible web content requires semantic information about them. Protocols and Formats Working Group of W3C WAI, created Accessible Rich Internet Applications (WAI-ARIA) report to define user interface elements, their roles and properties in accessible web pages. WAI-ARIA includes semantic role specification of these elements, accessibility states and properties and their requirements [38].

2.3 Segmentation

Segmentation has been used in many application areas; therefore, the term segmentation has several formal definitions. For example, in the field of image processing, segmentation refers to partition of a set of points of a picture into disjoint non-empty

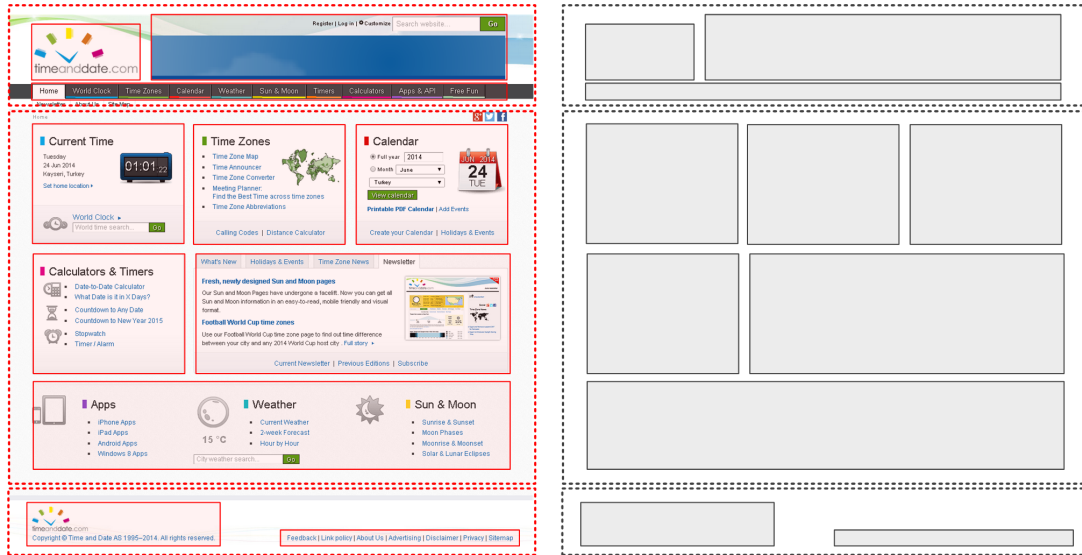


Figure 2.1: Sample segmentation of timeanddate.org page

subsets, in which, each subset is connected [110]. More specifically, web page segmentation refers to identification of adherent regions in a web page [113].

A segment is a part of the page with its unified subject, representing a visually continuous and cohesive area in the page when it is rendered [29]. Another widely used term to refer to segments is the term block. Web pages contain many blocks, which are shown in a rectangular region and can be located with width and length parameters [27]. Blocks have a hierarchical structure; as in visual elements, a block may contain smaller sub-blocks.

Figure 2.1 represents an example of block structure of Timeanddate.com page¹. It illustrates top two levels of segmentation process. In the first level, the page is divided into large visual blocks. Then, these visual blocks are further segmented to construct relatively smaller blocks. This process may also continue through deeper levels with respect to the granularity of web pages.

Segmentation is a natural process; when people perceive a web page, they unconsciously segment it with respect to some cues presented in the page. For example, people divide the web pages into visually separated areas [125]. Moreover, the web pages are designed in an organized way for improvement of readability. Therefore; web pages contain many cues for segmentation, including visual cues [131].

¹ <http://www.timeanddate.com/>, Last access: 24.06.2014

Segmentation has been used for various purposes. Informative and non-informative content blocks are separated with segments in web pages. This provides a useful source for web ranking and web data mining. Moreover, segmentation can be used to adapt web pages to small screen devices [29]. The effectiveness of web page segmentation has been proved in improvement of mobile browsing experience, information retrieval and web data mining [125].

2.4 Transcoding

Transcoding refers to the process of converting, distilling or transforming content or certain kind of data [69]. In the field of web accessibility, transcoding is a way of transforming inaccessible content into accessible form; so that, it becomes accessible for diverse devices, such as mobile devices, or diverse users, including visually disabled people [82, 7]. Therefore, the major intention of transcoding is to provide accessible content for disabled people [7] and mobile device users by transforming the content into more user friendly representation [84].

With the first efforts for improving web accessibility, transcoding was a major topic to provide accessible content for visually disabled users. When mobile devices became popular, transcoding gained a new function to adapt web content to these devices. After web pages became more visual and complex, transcoding came into prominence to make inaccessible content more accessible [7].

In order to solve the accessibility problems of users with different disabilities and the limitations of mobile devices, a various number of techniques have been deployed. These include text magnification, colour scheme changing, serialization, alternative text insertion, page rearrangement, simplification [7].

There are two main types of transcoding: manual transcoding and automatic transcoding. In manual transcoding, the designers and developers of the web page prepare an alternative version of the page. Many web pages, for example, provide a mobile mirror which is appropriate for mobile devices. In automatic transcoding, on the other hand, all web pages are automatically transcoded into accessible form by an external software. Both types of transcoding have some advantages and disadvantages.

Manual transcoding provides high-quality pages for a specific purpose [68, 115]; whereas, it is not applicable for all of the web pages [121, 115]. In contrast to manual techniques, automatic transcoding is adaptive for all of the pages and as a result, more cost-effective [121]. However, automatic transcoding quality is poor when it is compared with the pages which are transcoded manually [115]. Moreover, these techniques ignore the importance of different components in the web pages [68] and semantic information in the syntactic structure [115].

Transcoding architectures can be classified into three with respect to the publication and delivery model of the web pages: client-side transcoding, server-side transcoding and proxy-based transcoding.

In client-side transcoding, the whole content of the page is retrieved from the web server, and transformed locally in the device of the user [115]. It requires a specialized software such as a browser which is developed for this purpose or an extension on an existing browser. In this approach, user can specify a set of preferences for transcoding. However, it does not solve the limitations of the device which is used to access the web content. Moreover, users are required to install a new software to their devices. Furthermore, different software needs to be developed to fulfill the requirements of different device types and operating systems [2].

In server-side technique, the web pages are transcoded in the web server, and client software receives transcoded content. The main advantage of this approach is that, transcoding is handled in the original server of the web content; therefore, the cost of data transfer is relatively effective. Furthermore, this technique overwhelms the problems caused by the limitations of mobile devices. However, the transcoding operation has to be implemented in the web server of the content. Similar with manual transcoding, this approach is not applicable for all web pages [2].

Another transcoding approach is based on proxy servers. In this approach, the content is retrieved from web server, transcoded in a proxy server and then served to the client. In other words, the transcoding operation is independent of both content provider and the user. The main advantage of this approach is that, it is adaptive for all web pages. Moreover, it does not consume the resources of neither client-side nor server-side in transcoding operation. However, as a drawback, it puts an extra layer between client

and server, it increases the response time [2].

2.5 Ontologies

The term ontology originates from the Greek words *οντος* which means being and *λογος* which means theory or science. In philosophy, ontology questions what existence is and studies on categorization and organization of things which exist [95]. In Aristotle's perspective, an ontology defines a system which classifies the being [78]. The definition of ontologies in information science is relevant with its definition in philosophy. One of the people to define ontology was Gruber (1995), who wrote: "An ontology is an explicit specification of a conceptualization" [50]. Ontology is a basic form of knowledge representation about the real world to model a domain of knowledge. [78]

Ontologies can be classified into four: top-level ontologies, domain ontologies, task ontologies and application ontologies. Top-level ontologies describe very basic concepts which are not related with a certain domain or task. Domain ontologies, describe the things which are related to a specific domain, such as drugs. Similarly, task ontologies describe the concepts which are related to a task or an activity, for example diagnosing. Finally, application ontologies combine both domain and task ontologies to describe domain specific things while performing a specific task [51].

The main intention of the ontologies is to share common understanding of the knowledge about the meaning of things across different systems [101]. However, the aspects of the ontologies are not limited with this. One of the main reasons why ontologies are very useful in information systems is that, they organize the knowledge by providing access to it, optimizing it and simplifying reasoning. Therefore, they also provide mechanisms to use existing knowledge, as well as, manage alteration of it. Moreover, communication, integration and reasoning processes improve the functionality of the applications. Finally, by defining categories and relations between these categories formally, ontologies provide precise specification of the knowledge about things [112]. In conclusion, the key benefits of the ontologies are well structured organization of the knowledge, the communication between different systems and

improved functionality of the applications. These benefits ensured the popularity of ontologies in many applications.

The requirement of sharing common knowledge and describe the things on the web came with the need of a specification. In order to fulfill this requirement, Resource Description Framework (RDF) and RDF Schema (RDFS) were proposed. RDF is a simple assertion language which enables to define things with statements [49]. Although RDF is capable of defining properties of things, it does not describe semantic class definitions and subclass relations. RDFS extended RDF by providing a mechanism to define classes and subclass relations between them [15]. Besides these additional semantics, RDFS was also insufficient to describe the things completely, especially in requisite and sufficient requirements, cardinality restrictions, negation [66].

Filling these gaps, US and European research groups proposed DAML+OIL, which is a combination of DARPA Agent Markup Language (DAML) of the US Defense Advanced Research Projects Agency (DARPA) and Ontology Inference Layer (OIL) in 2000. This new ontology language came with cardinality constraints, ability define restrictions as well as necessary and sufficient conditions [88].

Finally, W3C revised DAML+OIL to propose Web Ontology Language (OWL), which extends RDF and RDFS in expressing semantics more exactly [89]. In 2009, W3C OWL Working Group released OWL 2, adding new functionality to express things better, such as property chains and enhanced data types and ranges [36].

Some of the features and characteristics of OWL can be listed as following [89]:

Class Classes define an abstraction mechanism for individuals which share similar properties. Example 2.5.1 illustrates the definition of `Person` class.

Example 2.5.1.

```
<owl:Class rdf:about="#Person" />
```

Class Hierarchies In OWL, classes can be defined as subclasses of one or more other classes. For example, `Male` class can be defined as a subclass of `Person` class as in Example 2.5.2.

Example 2.5.2.

```
<owl:Class rdf:about="#Male">
```

```

    <rdfs:label>Male</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>

```

Equality and Inequality OWL enables two classes to be defined as equivalent, in which, both classes have the same instances. This is used to define classes. For example, `Car` and `Automobile` are equivalent classes which have the same instances. Moreover, disjoint classes, which are ensured to have no individuals in common, can be defined. For instance, `Man` and `Woman` classes can be defined as disjoint classes, and do not share any individuals, as illustrated in Example 2.5.3.

Example 2.5.3.

```

<owl:Class rdf:about="#Car">
    <equivalentClass rdf:resource="#Automobile"/>
</owl:Class>

<owl:Class rdf:about="#Man">
    <owl:disjointWith rdf:resource="#Woman"/>
</owl:Class>

```

Object Property Object properties define relationships between individuals or individuals and data types. Example 2.5.4 illustrates the definition of `hasParent` object property.

Example 2.5.4.

```

<owl:ObjectProperty rdf:about="#hasParent"/>

```

Object Property Hierarchies Similar to classes, hierarchical features can be defined for object properties. An object property can be defined as a sub-property of another object property. For example, `hasMother` property in Example 2.5.5 can be defined as a sub-property of `hasParent` property. Moreover, synonymous properties can be stated to be equivalent in OWL. Finally, inverse properties can be defined. For instance, `hasMother` property can be defined as inverse of `isMotherOf` property.

Example 2.5.5.

```

<owl:ObjectProperty rdf:about="#hasMother">
    <rdfs:subPropertyOf
        rdf:resource="#hasParent"/>
    <owl:inverseOf rdf:resource="#isMotherOf"/>
</owl:ObjectProperty>

```

Domain and Range Domain and range of a property limit the individuals to which the property can be applied and which the property may have as its value. In Example 2.5.6, `hasMother` property must be applied to individuals in `Person` class, and must have individuals in `Female` class as its value.

Example 2.5.6.

```
<owl:ObjectProperty rdf:about="#hasMother">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Female"/>
</owl:ObjectProperty>
```

Object Property Restrictions Another feature of OWL is the restrictions. Restrictions state how properties can be used by the individuals. `allValuesFrom` restriction states a local range restriction on an object property which is associated with a particular class. Moreover, `someValuesFrom` defines that some values for the corresponding object property should be instance of the given class. Finally, `hasValue` states a certain individual as the value for the corresponding object property. Example 2.5.7 illustrates `allValuesFrom`, `someValuesFrom` and `hasValue` restrictions.

Example 2.5.7.

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasParent"/>
  <owl:allValuesFrom rdf:resource="#Person"/>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasParent"/>
  <owl:someValuesFrom
    rdf:resource="#Physician"/>
</owl:Restriction>
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasParent"/>
  <owl:hasValue rdf:resource="#John"/>
</owl:Restriction>
```

Individual The instances of the classes are individuals, which can be related with another individual with object properties. For example, John in Example 2.5.8 is an individual `Person` class whose mother is Mary.

Example 2.5.8.

```
<Person rdf:about="#John">
  <hasMother rdf:resource="#Mary"/>
</Person>
```

There are many tools and Application Programming Interfaces (API) to create, authorise, parse and reason on the ontologies. Ontology editors are very helpful in separating the structural domain and tool design, which generally implemented by knowledge engineers and detailed domain knowledge construction, which usually performed by domain experts. Protégé, for example, is a popular ontology authoring and editing tool [46]. Moreover, some software architectures may require to parse and render ontologies by using APIs. A notable example of API is OWL API. OWL API provides OWL validators, enables parsing the ontologies in different syntaxes including RDF/XML syntax and easily integrate with third party reasoners, such as HermiT [67]. Reasoners are utilised to query basic information in an ontology. This information may consist of subclasses of a particular class, its individuals and property values for an object property. HermiT, for example, is a well known ontology reasoner [105].

2.6 Rule Based Systems

A rule based system can be defined as a system in which premises infer conclusions by using rules based on knowledge to solve a particular problem. Rule based systems provide a mechanism which enables human experts specify a set of rules to solve a problem in terms of if-then conditions [59]. Being one of the most popular type of systems in artificial intelligent applications [40], rule based systems are robust tools for specifying knowledge and developing empirical applications [77].

Rule based systems are built on a production system, which is a rule based forward-chaining reasoning system. In production systems, general knowledge is represented in a definite form. A production system involves following main components [14]:

Facts Facts define basic knowledge and truths in a domain. A fact has a type, a set of attributes and their values in the following form:

$$(type\ attribute_1: value_1 \dots attribute_n: value_n)$$

Working Memory During the operation of a production system, assertions of facts are persisted in working memory component. Working memories are similar to databases, except that, it changes unceasingly as production system operates.

Production Rules Production rules represent reasonable conclusions and problem solving methods. A production rule consists of two main parts: a preceding set of conditions and a consequent set of actions. The conditions are satisfied with respect to the current state of the working memory, then consequent actions are performed, which modifies the working memory by creating new facts or altering the existing facts.

Rule engine is a software that interprets the rules and creates a working memory of facts. A notable example of rule engines is Jess, the Rule Engine for the Java Platform. Jess is a Java based rule engine and scripting environment, which uses an enhanced version of the Rete algorithm to process rules. Rete is a very efficient mechanism for solving the difficult many-to-many matching problems. Jess provides a simple mechanism to define structures, create facts and rules, and reason about them. Furthermore, Jess can also directly manipulate and reason about Java objects. Jess has a very simple syntax, which is similar to LISP and consists of a prefix function followed by a set of arguments. Basic concepts in Jess Language can be listed as following [45]:

Variables Variables can refer to a symbol, number, string or a list and created by using bind function. Variable names begin with question mark (?) character. The variables which are created with bind function, are cleared after the reset command. In order to create persistent global variables, defglobal function must be used. After reset command, global variables may assigned to their initial values or persist their current value, based on the reset preferences. Example 2.6.1 illustrates definition of both local and global variables.

Example 2.6.1.

```
(bind ?x 3)
(defglobal ?*y* = 5)
```

Working Memory A working memory is a collection of facts, and defined rules react to manipulation of these facts.

Templates Templates define the structure of a fact, similar to class definitions in Java. Each template consists of a name and a set of properties which are analogous to class members. In Jess, these properties are named as slots. A slot holds a

single value at a time. Multislots, on the other hand, can hold a list of values. Every fact in Jess has a template. In Example 2.6.2, the template definition for automobile facts is represented.

Example 2.6.2.

```
(deftemplate automobile
  "A specific car."
  (slot make)
  (slot model)
  (slot year (type INTEGER))
  (slot color (default white)))
```

Facts Jess provides definition of three different kinds of facts. First of all, unordered facts have well-formed structures which are defined with a template. Shadow facts, are similar to unordered facts, and used to map Java objects with facts in the working memory. Finally, ordered facts have simpler structure in a form of list. Example 2.6.3 presents the assertion of a sample fact to the working memory.

Example 2.6.3.

```
(assert
  (automobile
    (model LeBaron)
    (make Chrysler)
    (year 1997)))
```

Rules Jess rules consist of two main components. The first component, which appears in left hand side of the rule, specifies in which conditions the rule is fired. The other component at right hand side, defines the action taken when the rule is fired. In Example 2.6.4, a sample rule definition is illustrated.

Example 2.6.4.

```
(defrule car-classifier
  "Classify your car with respect to its
  production year"
  (automobile {year < 1980})
  =>
  (printout t "You have a very old car" crlf))
```

2.7 Eye Tracking

People make continuous eye movements when they read, look at or search for objects [99]. Eye tracking aims to measure these movements to interpret where an

individual is looking at in a particular time, as well as the sequence of the movements [97]. In the literature, eye tracking studies on web pages mainly focus on usability research [47] and understanding of user behaviour when they interact with a web page [20, 48].

To construct a common scanpath of visual elements based on eye tracking data, Eraslan et al. proposed eMINE Scanpath Analysis Algorithm [43]. The algorithm projects the fixations of eye tracking data over the visual elements and generates a path for each user. Then, the most common scanpath is calculated by using these individual paths in several iterative steps. 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. Finally, it returns the common scanpath of a particular web page. In this study, the results of this algorithm was used to simulate user behaviour of navigation in web pages.

2.8 Summary

This chapter has described the underlying technologies of web pages, web accessibility, processes to manipulate the structure of web pages and some concepts which can be used to describe semantic structure of a web page. The web is a very popular source of knowledge and various efforts has been made to improve the user experience. Although one of the major concerns of web and its contributors is to provide accessible web to anyone on a wide variety of devices, there has been some problems with web accessibility and web page adaptation for mobile devices. Although web accessibility concerns about all aspects of disability, in this study we focus on visual impairments. Visually disabled people use assistive technologies to be able to access web pages and W3C publishes some recommendations to ensure web accessibility. However, these recommendations are optional in the design and coding process of web pages.

The main problem with HTML documents is that, the language itself has a very flexible syntax. Even though HTML is XML-based, only syntactic regularity of the code is mandatory for proper rendering. The semantic structure of the web pages, on the

other hand, has less attention. Combining with the visual representation, which generally provides the semantic structure of the content, the designers who are inattentive or unaware of web accessibility standards may create inaccessible web pages. The main cause of this problem is that, it is possible to create the same visual representation by using different combinations of HTML tags and CSS styles. All of these representations may provide the same appearance; however, assistive technologies fall behind in providing the same functionality. This also prevents effective use of semantics of the tags, especially those were introduced with HTML5. Semantic definitions of visual elements require to be more comprehensive and benefit the aspects of visual representation along with the structural attributes, as well.

Web page segmentation is an important process to analyse the structure of web pages, since it provides a deep understanding of the structure, by dividing the page into smaller blocks. Moreover, once the semantic structure of a web page is properly understood, additional strategies can be developed to provide accessibility. Transcoding is one of the examples of these strategies, which enables to represent the page in alternative forms.

An ontology is a form of knowledge representation, which is capable of defining concepts, relations between them and their properties. The semantics of visual elements with their properties can be defined in an ontology. Thus, the knowledge of semantic description of web documents can be easily shared across different information systems. Furthermore, this knowledge can be represented in terms of formal rules, which can be used in a rule based system to reason on a web page to automatically detect its semantic structure.

In conclusion, this chapter reviewed fundamental web technologies, web accessibility and W3C standards to improve accessibility of web pages, segmentation, transcoding, ontologies and rule based systems. Moreover, we introduced eye tracking in brief. The next chapter describes the state of art in segmentation and transcoding explained in this chapter, as well as detection of heuristic roles of visual elements.

CHAPTER 3

RELATED WORK

The aim of this study is to propose a novel approach in automatic role detection and demonstrate its use in web page transcoding. Before moving on to the details of proposed system architecture, it is necessary to review literature surrounding web page segmentation, automatic role detection of visual elements and web page transcoding. These processes have been major tasks in many varied research fields. Therefore, numerous approaches have been proposed focusing on different aspect of web pages. In this chapter, we review the literature surrounding these approaches.

3.1 Web Page Segmentation

There is a large volume of published studies proposing different web page segmentation techniques. They all use different aspects of web pages. In general, web page segmentation algorithms can be divided into two with respect to their order of execution: top-down page segmentation and bottom-up page segmentation [5]. Top-down algorithms start with the root node of the page and segments the page through the leaf nodes [8, 25, 32, 58, 71]; while, bottom-up algorithms take leaf nodes at first and travel to the root node by examining different aspects of the nodes [29, 71]. In this section, we summarize different segmentation techniques proposed in the literature.

3.1.1 HTML Tag Based Approaches

One main criterion used in web page segmentation is the types of HTML tags. Lin and Ho [79] assume that 70% of the pages are created with `table` elements; hence, they concentrate only on the pages which contain `table` element. They segment the pages with respect to `table` tag and a narrow set of additional tags. Bar-Yossef and Rajagopalan [9] define blocks as pagelets, which are HTML elements whose children do not contain no more links than a threshold and ancestors are not pagelet. Therefore, this approach is mainly based on `a` tags, which create hyperlinks in the web page. Xiang and Shi [118] consider a web page as a composition of basic blocks and separators. Therefore, they focus on detecting content and separators in the page. The separators are detected with respect to `table` and `hr` elements. Xiao et al. [122] segment web pages with respect to `table` tags. It is a DOM based approach and starts from the leaf nodes of DOM tree. Until there is no remaining node suitable for merging, the algorithm merges the nodes iteratively with respect to a set of attributes of the nodes.

3.1.2 Position and Alignment Based Approaches

Some approaches also use position and alignment, as well as the distance between the visual elements. Kovacevic et al. [72] take DOM structure of the web pages and create a virtual screen that corresponds to a coordinate system. This coordinate system specifies the positions of the visual elements in the page. Using a predefined set of heuristics, they segment the page based on the *x* and *y* axes of visual elements on the coordinate system. Yin and Lee [130] propose a DOM based segmentation algorithm, which calculates a likelihood between two nodes to check whether two nodes belong to the same block. The likelihood value is calculated with respect to a set of similarity metrics and distance between two nodes. Similarity metrics consist of conditions including whether two nodes are vertically or horizontally aligned, next to one another, and have the same tag or parent node. Guo et al. [52] assume that relevant content in the page layout tends to appear near each other and share the same alignment. In order to segment web pages, they calculate the coordinates of each visual element along *x* and *y* axes and iteratively try to find consistency in the

elements. Consistency is obtained based on alignment of elements in either x or y axes. Then, they try to determine a similarity between blocks, based on visual styles, such as font size and font color. Borodin et al. [86] and Mahmud et al. [13, 85] assume that semantically related content is often represented as aligned in web pages. By using a geometrical clustering algorithm, they segment web pages into meaningful blocks. Whether a visual element should be divided is decided based consistency examination. If the children of a visual element are horizontally or vertically, then the visual element is considered as consistent. Burget and Rudolfova [18, 17, 19] use bottom-up box clustering approach to segment web pages. First, they create a tree of boxes, which contains x and y coordinates, width and height of the visual elements in the page. Then, they segment the pages with respect to whether a visual element directly contains text, background color of the visual element is different from the background color of its parent and it is separated by a visual frame in at least one of its borders. Cheng and Gotz [58] and Hattori et al. [34] calculate the distance between nodes in DOM structure and segment the web pages by comparing the distance to a threshold value. In this approach, rather than absolute position of the visual elements in the page, definition order and depth are used as a distance metric.

3.1.3 Size Based Approaches

Some algorithms segment the web pages based on the size property. Chen et al. [32, 33] segment a web page in two main steps. First, they start with DOM tree of the page and try to classify nodes with respect to their width and height. If width or height of a node is larger than specific thresholds, then the node is divided further. Then, they utilize two methods to further divide the blocks. The first method is explicit separator detection, in which three types of separators are used, which are `hr` tags, border attributes of `table` and `div` elements and images. The other method is implicit separator detection, which aims to detect blank areas intentionally created by the author. Gupta et al. [53] take DOM structure of the web page and calculate the width and height of nodes. They segment the page until they reach a node which properly fits in a specific area. This area is intended to be the screen size of a mobile device. Vineel [113] focuses on node content size and node entropy for each visual element in the page. If the size of a visual element is larger than a threshold, then it is further

divided. In order to calculate node entropy, specific patterns are detected. These patterns are based on the frequency of tags which appear in the DOM structure of the page. It is assumed that, certain tags appear in a certain depth in the DOM tree. Based on this assumption, he calculates a strength value for regular patterns as node entropy. If the node size and entropy satisfy a threshold, then the node is no longer divided. Sano et al. [102] assume that web pages are structured based on a set of different layouts. First, they compare the size of the nodes with a threshold to determine if the nodes should be divided. Then, they detect the layout of the page among a predefined set of layout options. Finally, they detect title blocks and group relevant blocks with corresponding title blocks.

3.1.4 Vision Based Approaches

Visual representation of visual elements have been used in some methods. Chen et al. [30] assume that relevant objects have similar visual styles and there exist boundaries between irrelevant objects. First, they render the visual image of the page and apply a visual similarity pattern detection method by analyzing object properties. Yi et al. [128] take the DOM structure of the web page and create a style tree. Then, in a top-down fashion, they check whether the visual styles of a node is the same with the styles of its parent. In case of dissimilarity between the styles of a node and its parent, then the parent node is segmented. Visually same nodes, on the other hand, are merged to construct blocks. Cai et al. [25, 24, 131] and Yu et al. [26] propose the Vision Based Page Segmentation (VIPS) algorithm. VIPS segments web pages by using both the DOM structure and the visual presentation of the page. First it divides the visual elements to blocks based on a set of rules related with visual styles of elements. Then, it detects the separators, which separate the blocks from each other. Finally, it builds a tree structure of the blocks extracted. These three steps are iteratively repeated until there is no visual element to examine. VIPS has been a very popular segmentation algorithm and used in many other studies. In brief, VIPS has been used in identification of visual element importance [107], web searching [22, 24, 25, 26, 131], image searching [21, 23, 60, 114] and visual quality evaluation [117].

3.1.5 Machine Learning Approaches

Machine learning is another approach, used in web page segmentation field. Baluja [8] proposes a decision tree based web page segmentation algorithm. Based on the cues from the DOM structure of the page, he applies computer vision algorithms and try to classify each visual element in the page recursively. Moreover, Chakrabarti et al. [29] propose a machine learning algorithm to segment web pages, based on a combinatorial optimization framework. They represent DOM nodes in a weighted graph, which is used in two concrete instantiations: correlation clustering and energy-minimizing cuts in graphs. Using hierarchical structure, location and visual layout of the nodes, it is determined whether a subset of nodes should appear in the same block or not. Alcic and Conrad [5] also aim to segment web pages based on clustering by using DOM based, geometrical and semantic features of visual elements. A distance measure is used to group coherent content.

3.1.6 Repeattion Based Approaches

Some approaches also use repetitive patterns in the web page. Hwang et al. [115] and Whang et al. [68] assume that multiple repeated layout patterns are used in complex pages in general. Mainly, they convert HTML tags to a pattern, represented as string. Using this string, they look for a repeated pattern. They segment the page by grouping the visual elements based on the repetitive patterns. Nanno et al. [92] focus on detecting repetition structures by deploying some heuristic rules. Kang et al. [70] also presume that a visual element has semantically similar children elements. They use a repetition based algorithm to segment pages. First, they list children nodes in sequential order and detect repetitive tag occurrences. Then, using these repetitive patterns, they use them to segment the web pages.

3.1.7 Custom Approaches

Finally, some algorithms use custom techniques to segment pages. Kohlschütter and Nejd [71] propose a web page segmentation algorithm based on the text density. They compare the similarity of text density of two adjacent nodes. If the similarity

value is under a threshold, they represent the nodes as two different blocks. Cao et al. [27] assume that, web pages are constituted by blocks and these blocks are separated by background space. It means that, if there is no background space between two visual elements, they belong to the same block. First, they convert the web page to image and apply image processing to detect blocks. Xiang et al. [119] and Yang and Shi [125] use Gestalt Theory for web page segmentation. According to the Gestalt Theory, four general laws guide human perceptive process. These rules are proximity, similarity, closure and simplicity, which suggest precepts about organization of similar or related items. Based on these rules, they aim to segment web pages by detecting similar or relevant visual elements.

3.2 Heuristic Role Detection of Visual Elements

In the previous section, we reviewed the literature surrounding visual element identification. The methods previously presented only focus on identification of visual elements in web pages without their corresponding roles. This section moves on to review techniques to discover roles of visual elements in web pages. There are different approaches for automatically identifying visual elements and discovering their roles [25]. Our literature review showed that, in different fields, role identification has been proposed to serve different purpose.

Role identification has been used to adapt web pages to small screen devices. Yin and Lee [130] propose five different roles, which include content, related links, navigation and support, advertisement and form. They detect the roles of visual elements based on random walks on a directed graph. Similarly, Ahmadi and Kong [3] define five roles: top, main content, left and right menus, bottom and clutter. In this approach, role of a block is assigned with respect to its position. Chen et al. [33] also propose four different roles, which are header, footer, sidebars (left and right) and the body. The role detection is applied based on basic assumptions about a template based page structure. Finally, Xiao et al. [123] propose a simple method which focuses only on link blocks and content blocks. On the other hand, Chen et al. [30] propose Function-Based Object Model, which is a very detailed model, that applies a set of rules on the visual elements and a set of their properties.

Role detection has also been proposed for information retrieval and web data mining applications. Kovacevic et al. [72] define a set of roles which indicate common areas in web pages, including header, footer, left and right menus, and the center; and apply a set of heuristics to detect the roles for guiding page classification. Lin and Ho [79] also classify visual elements as informative content blocks, which include significant content to users and redundant content blocks, which include navigation or advertisement panels. An entropy value is calculated by using the tokens in the content of the visual element and compared to a threshold to determine the class of the visual element. Similarly, Burget and Rudolfova [19] aim to identify a set of roles which include main article, heading, subtitle, paragraph, data, author and author date, by using a decision tree classifier on visual features. Liu et al. [80] propose three types of roles, navigation bar, navigation list and content block, based on the number of links in the visual elements. Finally, Yi et al. [128] aim to differentiate noisy blocks and main content, by examining a set of visual representation styles.

Web accessibility and interface improvement are other fields where role identification has been utilized. Takagi et al. [108] propose different roles for visual elements which include: proper content, updated index, general index, norole, header, footer, advertisement, delete and layout table. In this approach, the visual elements in web pages are manually annotated, so that, an annotation matching mechanism detects the corresponding roles of visual elements. Xiang and Shi [118] define two types of visual elements: non-text blocks and text blocks based on whether a visual element is a leaf node or not. If a visual element is a leaf node, except for the elements which directly contain text, then the visual element labeled as non-text block. For text blocks, on the other hand, further properties are examined to label visual element as text block.

There has also been considerable effort to construct knowledge bases and standards to semantically describe visual elements. For example, Web Authoring for Accessibility (WAfA) Ontology [127] provides description of roles of visual elements in a rich vocabulary. WAI-ARIA [38] also provide a role taxonomy and define attributes of these roles. Finally, HTML5 introduces new tags which enable to specify semantics of visual elements.

3.3 Web Page Transcoding

Over the past decade, various web page transcoding approaches have been proposed to improve web accessibility. These approaches can be analyzed in two main categories. Some transcoding approaches aim to improve web accessibility for disabled people; while others focus on improvement of web accessibility in mobile devices. In this section, we review each of these categories separately.

3.3.1 Web Accessibility

A considerable amount of literature has been published in the field of web accessibility, which utilize manual annotation of web pages. Asakawa and Takagi [6] and Takagi et al. [108] propose a method which rearranges the layout of the web pages by moving more significant segments to the upper side of the page. The significance of a segment is determined with respect to manual annotation of segments with a pre-defined set of roles, including header and footer. The aim of this study is to make important parts of a web page more easily accessible by using screen readers. Furthermore, Yesilada et al. [127] transcode web pages to improve their accessibility based on an ontology named WAFa. WAFa provides description of roles of visual elements in a rich vocabulary [56]. Harper et al. [82] and Lunn et al. [55, 83], on the other hand, annotates the CSS attributes by using a simple set of vocabulary. However, these approaches would appear to be inefficient in automatic transcoding of web pages.

Borodin et al. [86] and Mahmud et al. [13, 85] draw our attention to the elapsed time which takes to access a particular part of a web page by using assistive applications. According to them, the main reason why browsing in assistive technologies is time-consuming is that, they follow the DOM structure of the page in sequential order. To solve this problem, they propose a method which skips to the significant content of the page by using a geometrical clustering algorithm to identify segments in a web page automatically.

3.3.2 Small Screen Adaptation

In recent years, there has been an increasing number of studies that aim to adapt web pages for devices with small screens. Some of these studies are based on removal of unnecessary or irrelevant content from the web page. Chen et al. [30] identify the visual elements which are used for decorative or special purposes, such as advertisements, and remove them from the web page layout. Similarly, Xiao et al. [122] transcode a web page by removing irrelevant segments in the page. One major drawback of this approach is that remaining relevant or significant segments may not yet properly fit in the screen of the device. Another problem with this approach is that, it prevents users from accessing removed content on purpose; hence, it causes some parts of the page to be inaccessible. Some approaches, on the other hand, are based on selecting a specific part of the page and displaying only this particular part to the user. Yin and Lee [129] aim to detect the important parts of the page and display only these parts on the screen. Similarly, Xie et al. [124] and Yin and Lee [130] propose a method which identifies the most important part of a web page and only display this part. Although these approaches may provide a simplified version of the page and ability to access important parts easily, the parts identified as unimportant may also be important for some users. In this case, these parts become inaccessible for such users. Unlike these approaches, Song et al. [107] sort the visual elements with respect to their importance and provide a mechanism that enables sequential access to visual elements in order of importance.

Other approaches summarize the web page and display only a specific part of the page based on the users' selections. Ahmadi and Kong [3] and Hattori et al. [58] transcodes web pages by creating a summary of the page in form of a table of contents. Chen et al. [32, 33] also segment the web pages and connect the segments to one another by using links. Some methods achieve summarization with alternative representation of visual elements. Xiao et al. [120] propose a model named tap and display model. In this model, overall page is presented and after users select a particular part, content of the corresponding part is displayed. Baluja [8]; Chen et al. [32, 33]; Xiang et al. [119]; Xiao et al. [123] and Yang and Shi [125] propose a thumbnail view of the web pages. In this view, a summary of the web page is provided with thumbnails. Users select a

thumbnail to access corresponding content. Moreover, Xie et al. [124] improve this approach by providing a highlighting method based on importance of visual elements. Another possible technique is to summarize the visual elements without changing the page layout. Whang et al. [115] and Hwang et al. [68] keep the original page layout by converting titles of each element to a link. The links point to the content of the corresponding visual element. Similarly, Hwang et al. [68] construct smaller subpages by segmenting the web page. Then, they bind these subpages by providing some hyperlinks in sequential order. Ahmadi and Kong [3] represent only the first sentence of each visual element.

Some web pages contain media objects such as an embedded video, which in some cases may not fit in the screen. To solve this problem, these visual elements can be scaled down with respect to screen size. Moreover, altering the presentation of modality of an object can also provide a solution [2]. However, our main purpose in this study is to present visual elements in their original formats.

3.4 Summary

This chapter has described the methods and approaches used in web page segmentation, identification of visual elements and web page transcoding. Together these methods provide important insights into automatic identification of visual elements in web pages.

There are many studies which aim to segment web pages into meaningful blocks. web page segmentation has been utilized in many different areas, and different studies focused on varied aspects of web pages to segment pages. However, the generalisability of much published research on segmentation field is problematic. For example, some approaches are based on strict assumptions and some of them examines only a narrow set of attributes of visual elements. VIPS, being one of the most popular segmentation algorithms, provides a comprehensive mechanism, since it takes both structural and representational features of the web pages. However, VIPS also has some limitations, such as, lack of HTML5 features and ambiguity in rules. Therefore, an enhanced version of VIPS is required.

Identification of roles of visual elements has been a major task in many fields including adapting web pages for small screen devices, creating intelligent web interfaces, supporting information retrieval and data mining on the web, and improving web accessibility. Although considerable amount of research conducted to identify visual elements, much of them up to now focus on a narrow set of roles of visual elements and their features. WafA, WAI-ARIA and HTML5 supplies good resources of knowledge about visual elements; however, they still do not cover a deep understanding of visual elements and their features.

Finally, many studies propose transcoding approaches to adapt web pages for disabled people and mobile devices. Although they propose different and useful strategies to transcode web pages, none of them comprehends neither the heuristic roles of visual elements in detail, nor the user behaviour with eye tracking experiments.

In conclusion, although many techniques have been proposed to segment web pages, identify visual elements in these pages and transcode them to alternative representations, these techniques have many limitations. The next chapter describes our proposed approach and system architecture to solve these limitations.

CHAPTER 4

AUTOMATIC IDENTIFICATION OF VISUAL ELEMENTS IN WEB PAGES

In this chapter, we present our novel approach to automatically detect the heuristic roles of visual elements in web pages. Our approach mainly consists of identification of visual elements, knowledge description and formalization for roles and their characteristics, and automatic detection of the roles of visual elements based on constructed knowledge.

This chapter first gives a brief overview of overall system architecture in Section 4.1. Section 4.2 explains visual element identification process with our improvements on an existing segmentation algorithm. Then, Section 4.3 explains our novel approach to detect heuristic roles of visual elements in web pages. Finally, Section 4.4 presents the implementation of the system.

4.1 System Architecture Overview

The overall architecture of the proposed system has two main components, which can be summarized as following:

Visual element identification: The main purpose of this component is to parse underlying source code of a web page to construct a tree of visual elements, by segmenting the page with respect to its visual representation and DOM structure.

Automatic role detection: This component mainly converts the ontological knowl-

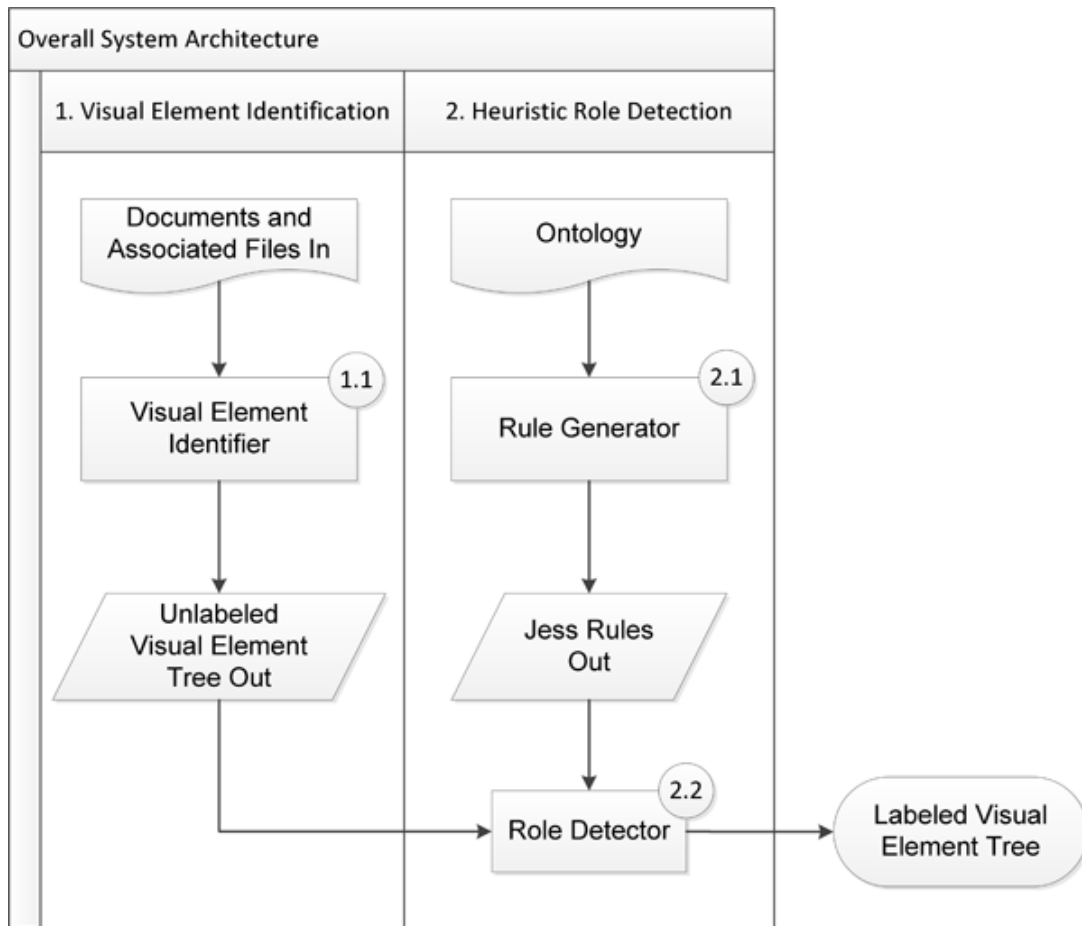


Figure 4.1: System Architecture

edge to heuristic rules in an appropriate syntax. Then, it applies these rules on the tree of visual elements, to automatically identify the corresponding role of each visual element in the web page.

The overall architecture of our proposed system is illustrated in Figure 4.1. Visual element identifier (indicated by ①.1), segments the web page to construct a tree of visual elements. The rule generator (indicated by ②.1) converts the formal ontological knowledge to a set of rules. Finally, role detector (indicated by ②.2) receives the tree of visual elements and applies the rules on each visual element to determine its role. The components execute in sequential order; so that, the first component constructs a tree of visual elements and pass it to the automatic role detector. The second component, then, applies generated rules on the visual elements and returns labeled tree of visual elements.

During the development of our system architecture, we took the limitations of previous methods into consideration. Based on these limitations, the design criteria of our approach can be discussed under three headings: modifiability, maintainability and coverage.

Modifiability: Previous role detection techniques are based on static definition of roles and their characteristics. Consequently, these approaches can be used only in a specific field. Our approach is based on a modifiable knowledge base, so that, it can be modified with respect to application specific requirements. This includes both extension of role set and characteristics to describe these roles.

Maintainability: Web trends and technologies change and improve rapidly. Static definitions of roles and heuristics fail to adapt the role detection method to new trends and technologies. Modifiable knowledge base enables reconstruction of role set and heuristics easily for future applications.

Coverage: Role detection approaches in the literature do not provide neither comprehensive understanding of the roles of visual elements nor describe their characteristics in detail. In our approach, we analyzed wide range of roles and visual element attributes to create a comprehensive knowledge base.

4.2 Visual Element Identification

Web pages typically consist of many visual elements. The main purpose of the visual element identifier component is to identify these visual elements in the page by conducting a segmentation process.

Our study started with searching for a basis to identify visual elements in web pages. As was mentioned in Chapter 3, a variety of approaches have been proposed to segment web pages. Based on their segmentation criteria, these approaches can be grouped into two: DOM based approaches and vision based approaches. DOM based approaches mainly use HTML tags of visual elements or repetitive patterns in DOM structure. Vision based approaches are generally based on the DOM structure; on

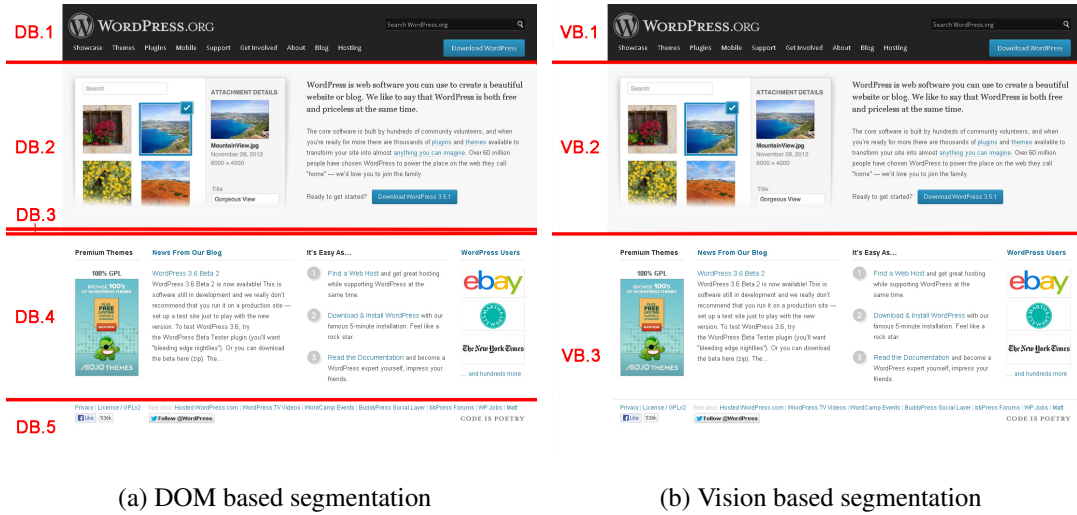


Figure 4.2: Comparison of DOM based and vision based segmentation

the other hand, they also take a set of visual attributes of the visual elements. DOM structure of a web page provides significant information about structural association between its nodes; while, visual styles cannot describe the structural relations on their own. However, they provide substantial cues about semantic relations between visual elements. For example, the space surrounding a visual element may imply that, it is irrelevant to adjacent visual elements. DOM, on the other hand, do not have such capability. Therefore, vision based segmentation approaches appear to be more comprehensive in using different aspects of the web pages.

Figure 4.2 represents a comparison between DOM based segmentation and vision based segmentation. As can be seen in this figure, vision based segmentation process (See Figure 4.2b) extract simpler set of visual elements than DOM based segmentation in Figure 4.2a. There are several possible explanations for this. First of all, DOM structure on its own cannot detect invisible nodes, which do not enclose meaningful content in overall page. Vision based segmentation can achieve this by examining a set of attributes of a visual element. Moreover, some visual styles provide ability to distinguish irrelevant content or to group some bulk content with creating similar representation. For example, two adjacent visual elements with different background colors can be treated as separate visual elements in general. However, DOM based segmentation lacks the ability to conduct such investigations.

It appears that, both DOM structure and visual representation of web pages supply

significant cues for segmentation process. Therefore, an optimal method should enclose both aspects of the web pages in combination. One well-known algorithm that both utilizes DOM structure and visual styles is VIPS. In this study, VIPS is mainly used to identify visual elements in the web page structure. However, VIPS also has some limitations, which are discussed in the following section.

4.2.1 VIPS Algorithm and Its Limitations

VIPS is a segmentation algorithm which combines the utilities of both the DOM structure of a web page and its visual representation. The main idea of this algorithm is that, web pages are composed of visual elements and these elements can be identified by using a set of visual and structural cues. As discussed in Chapter 2, web pages are created by using HTML tags, and visual styles of HTML elements are specified with CSS. VIPS therefore parses underlying HTML and CSS codes to retrieve the DOM structure of a web page and its visual attributes. It is a top-down approach, so that, it starts from the root of the underlying DOM tree and iteratively segments the page through the leaf nodes.

The original algorithm has three main steps: visual block extraction, separator detection and content structure construction. In visual block extraction, first DOM nodes are categorized based on their appearance on the browser and children nodes into following classes:

Inline node According to W3C HTML4 Specification [98], an inline node does not begin with a new line and contains only data or a set of inline nodes.

Line break node Line break nodes are referred as block-level elements in W3C HTML4 Specification and begin with a new line. Line break nodes are large block structure which can contain both inline and line break nodes.

Valid node A node is valid if it can be visually perceived in the browser.

Text node A text node only contains free text without any additional HTML tag.

Virtual text node A virtual text node is either an inline node which only contains text node or a composition of text nodes and virtual text nodes.

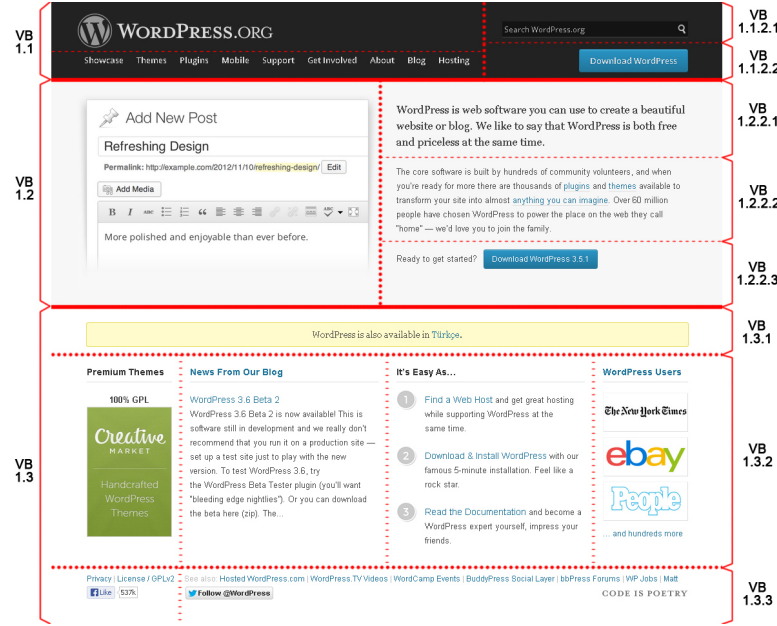
After node classification, a set of heuristic rules are applied on the nodes to determine whether corresponding node should be segmented further or represented as a single element in the visual element tree. It is decided based on the structural and visual attributes of the nodes. These attributes include HTML tag of the node, comparison of its background color with the parent node, the types of its children nodes and its size.

The following step, separator detection, aims to detect the separators between the visual element, which were extracted in the previous step, and then weight them with respect to some attributes of visual elements surrounding the separator. The weight of a separator indicates the relation of these visual elements. Finally, extracted visual elements are placed in the visual element tree of the web page. Until extracted elements satisfy a granularity requirement or a leaf node is reached, these three steps repeat iteratively [25].

The algorithm initially takes underlying source code of a web page and produces a hierarchical tree of visual elements at the end. It should be noted that, this tree of visual elements is not identical to DOM tree of the page, which can be explained with a few reasons. First, in visual block extraction process, invalid nodes, which appear in the DOM but not visually appear on the browser, are eliminated. Furthermore, adjacent text nodes and virtual text nodes are grouped into a single visual element to provide semantic coherence. Finally, after a granularity requirement satisfied in a visual element, VIPS stops segmenting it, meaning that, it is not guaranteed that all leaf nodes are reached in segmentation process.

Figure 4.3 demonstrates the result of segmentation process for Wordpress.org page both visually and in tree based notation. First, VIPS segments the page into larger visual elements, such as VB1.1, VB1.2 and VB1.3 in Figure 4.3a. Then, segmentation iterates until a granularity requirement is met or a leaf node is reached. For example, in the next iterative step, VB1.3 is further segmented to VB1.3.1, VB1.3.2 and VB1.3.3. Figure 4.3b represents the tree structure of the tree of visual elements with corresponding XPATH for each visual element. It can be seen from this figure that, some nodes in DOM tree do not have corresponding visual elements in the tree.

VIPS effectively combines both tag attributes of the nodes in DOM structure and their



(a) Visual representation

Block Name	XPATH
VB1	/HTML/BODY/
↪ VB1.1	/HTML/BODY/DIV/
↪ VB1.1.1	/HTML/BODY/DIV/COMPOSITE/
↪ VB1.1.1.1	/HTML/BODY/DIV/DIV/H1/
↪ VB1.1.1.2	/HTML/BODY/DIV/DIV/UL/
↪ VB1.1.2	/HTML/BODY/DIV/COMPOSITE/
↪ VB1.1.2.1	/HTML/BODY/DIV/DIV/DIV/
↪ VB1.1.2.2	/HTML/BODY/DIV/DIV/DIV[2]/
↪ VB1.2	/HTML/BODY/DIV[2]/
↪ VB1.2.1	/HTML/BODY/DIV[2]/DIV/IMG/
↪ VB1.2.2	/HTML/BODY/DIV[2]/DIV/COMPOSITE/
↪ VB1.2.2.1	/HTML/BODY/DIV[2]/DIV/P/
↪ VB1.2.2.2	/HTML/BODY/DIV[2]/DIV/P[2]/
↪ VB1.2.2.3	/HTML/BODY/DIV[2]/DIV/P[3]/
↪ VB1.3	/HTML/BODY/COMPOSITE/
↪ VB1.3.1	/HTML/BODY/DIV[3]/
↪ VB1.3.2	/HTML/BODY/DIV[4]/
↪ VB1.3.2.1	/HTML/BODY/DIV[4]/DIV/DIV/
↪ VB1.3.2.2	/HTML/BODY/DIV[4]/DIV/DIV[2]/
↪ VB1.3.2.3	/HTML/BODY/DIV[4]/DIV/DIV[3]/
↪ VB1.3.2.4	/HTML/BODY/DIV[4]/DIV/DIV[4]/
↪ VB1.3.3	/HTML/BODY/DIV[5]/
↪ VB1.3.3.1	/HTML/BODY/DIV[5]/COMPOSITE/
↪ VB1.3.3.2	/HTML/BODY/DIV[5]/COMPOSITE/

(b) Tree structure notation

Figure 4.3: Visual elements and tree structure for Wordpress.org

visual styles to identify meaningful visual elements in the page. However, when we try to implement and integrate VIPS to our system, we found that it has some limitations. These limitations can be grouped into two: corrective limitations and adaptive limitations. Corrective limitations are related with rule definitions and node categories. Adaptive limitations are caused by its incompatibility with new technologies, such as HTML5 and dynamic web content.

Our investigation of VIPS showed that, the rule definitions and node categorization do not fit in real web pages. First of all, it proposes a set of rules to determine whether a node represents a coherent visual element on its own, or it should be further segmented into smaller visual elements; however, the precedence of the rules are not properly defined. This results in ambiguities when choosing the appropriate inspection. Moreover, some rules compare the size of a node to a threshold which varies in different cases. Nevertheless, definition criteria of such values are not covered in the algorithm. Also, the rules vary for different HTML tags of the nodes and only inline text nodes, `table`, `tr`, `td` and `p` tags are specified; while, the other HTML tags are involved within the same group. However, we observed that, some HTML tags introduce extra space around their borders than the others. Therefore, further classification of the HTML tags is required.

The most recent version of HTML, HTML5, was proposed after VIPS and extended HTML tag set. Therefore, HTML tags defined in the original algorithm do not include those which are proposed with HTML5. This is an important limitation for implementation of the system to cover modern web pages. Moreover, modern web pages also contain dynamic visual elements and the structure and the content of such elements are updated without reloading the overall page in predefined intervals or with respect to user interaction. When we analyze these elements, we saw that these elements are invisible at the initial state of the page, and after an event is triggered, they become visible on the screen. However, VIPS classifies them as invalid and do not catch the events affecting their visibility. Rather than applying on the page in a certain state, a mechanism which traces the dynamic changes in the content is also required.

In order to use VIPS algorithm in our system, we have made additions to the original

algorithm. These additions include reconstruction of the rule set, extension of the tag set and further classification of the node types. The next section explains our enhancement and implementation process.

4.2.2 VIPS Improvements

At the beginning of the development of visual element identifier component, all HTML and HTML5 tags were classified under three categories: invalid tags, inline tags and line break tags. Corresponding category of a tag is determined with respect to whether it appears on the browser and introduce a new line, without including additional CSS styles. Overall tags and their categories are presented in Appendix A.

Another enhancement in implementation process was related with the classification of the line break nodes. It was observed that, different block-level HTML tags introduce various amount of margins around or paddings inside their borders. For example, an observation on a set of browsers including Google Chrome v35, Mozilla Firefox v30 and Internet Explorer 11 showed that, a `p` tag introduce 16 px margin only at the top and bottom of its borders, even if no additional styles are defined. A `div` tag on the other hand, has 8 px margin at its all sides. These differences emphasize the distinct behaviours of different tags, and this can be used to separate irrelevant content in the page layout. However, `margin` and `padding` properties of visual elements can change with respect to underlying styles, therefore, examination of `margin` and `padding` property values provide a deeper understanding about a visual element. In contrast to line break node classification, `div` tag is handled as a special case in our implementation, since its main intent is to provide a grouping visual element for the others. Finally, `float` property was included in the visual cue set. This property helps to identify irrelevant content groups under a visual element if some of the children elements have different `float` property value than the others. In conclusion, some additional style properties were included in the visual cues during the implementation of the algorithm.

Apart from the underlying tag, a node can be invisible and hence invalid, based on some of its attributes. We have seen that, many designers adopt different techniques to change a visual element to invisible. Some of these techniques can be listed as

following:

display property Setting `display` property of a visual element to `none` makes it invisible.

visibility property Similar to `display` property, a visual element is invisible if its `visibility` property is set to `hidden`, but in this case, the visual element still takes up space in the layout.

position property `fixed` and `absolute` position properties with negative `left` and `top` properties may cause a visual element to fall out of the page border on the browser and make the visual element invisible on the screen.

text-indent property Negative `text-indent` property values may also be used to make a visual element invisible.

Size Visual elements with width and height of less than 1 px are handled as invalid nodes in our implementation, since they do not represent valid content.

Content Some leaf nodes which do not have textual content require further examination. Such cases should be handled with respect to the tag type. Line break leaf nodes without textual content produce additional space between valid visual elements; therefore, such cases should not be classified as invalid. On the other hand, inline leaf nodes without text child has no valid effect on the visual representation; hence, they should be handled as invalid.

Finally, it was observed that, web pages typically have the layout which is represented in Figure 4.4. In some cases, high level visual elements indicated by ①, ②, ③ and ④ appear in the same level as sibling nodes in DOM tree. However, when compared on another, it was found out that, ② and ③ are more relevant, and it was required to separate them from ① and ④. In order to handle such cases, a comparison of visual element sizes were conducted and visual elements like ② and ③ were grouped under a visual element. These parent visual elements do not have the counterpart nodes in DOM structure and regarded as composite elements in our implementation.

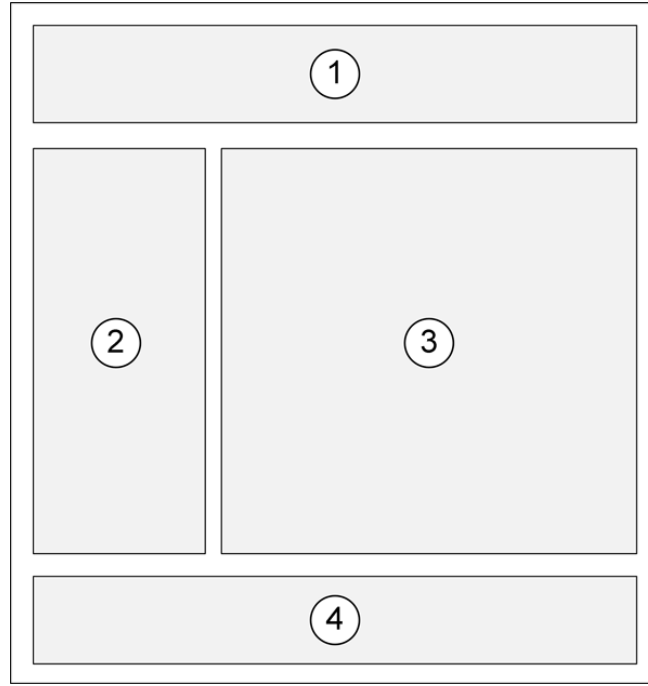


Figure 4.4: A typical layout for web pages

Before implementing basic steps of VIPS, we included a DOM structure construction step to determine the nodes in DOM tree and their properties. Unlike original VIPS algorithm, which goes down through a certain level in DOM, our mechanism traverses the overall DOM tree in a top-down fashion. Although it increases response time and memory usage when it is compared to VIPS, it has the advantage of fully examining overall web page structure and its content for later use in role detection. However, for certain applications that require high performance, our infrastructure enables to traverse until a granularity requirement is satisfied or a certain level is reached in DOM tree. DOM structure construction process produces a temporary hierarchical tree of visual elements similar to DOM tree, except that invalid nodes are extracted.

After DOM structure construction ends, visual block construction begins. This part is very similar to VIPS; however, a different set of rules is used. By considering enhancements explained above, a new rule set was defined to apply on visual elements. This new rule set includes some rules from the original algorithm, as well as new rules which we defined. Appendix B presents the rule set used in visual element identification. The precedence of these rules are designated with their definition order. Similar to VIPS, our visual element identifier also runs iteratively from the root node

through leaf nodes. After visual elements are extracted, the content structure of corresponding visual element is constructed. In our implementation, separator detection process is omitted, since the relation between visual elements are determined with a set of heuristics in role detection component.

At the end of visual element identification, a hierarchical tree of visual elements is created. This tree of visual elements is then used in the automatic role detection component. The following section describes the automatic role detection mechanism for visual elements in the tree.

4.3 Automatic Role Detection

After explaining the visual element identification component, this section focuses on our novel approach for automatic role detection of visual elements in web pages. The aim of this component is to automatically detect the corresponding role of each visual element in the tree of visual elements, which is constructed by the first component.

As was mentioned in Chapter 3, current approaches focus on a set of attributes of visual elements to detect their roles. However, these attributes are very limited and do not comprehend all aspects of visual elements. Moreover, these approaches are based on static rules and cannot be modified for different requirements. Another limitation of previous methods is that, HTML and CSS have flexible nature which enables to create the same visual representation with alternative coding. Hence, it is not possible to strictly define sufficient criteria to be successfully applied on visual elements in every web page for a certain role.

Consider the two title elements represented in Figure 4.5. The visual element in Figure 4.5a is created by using a `h1` tag, and the source code is given in Figure 4.5b. The other visual element, which is represented in Figure 4.5c, is created with a combination of `p` tag and some style attributes, given in Figure 4.5d. These two visual elements are visually identical and may properly represent title elements in visual representation. However, when it comes to web accessibility, assistive technologies cannot identify the visual element in Figure 4.5c as title. This simple example shows that, strict rules may fail often in determining the role of a visual element.

A Sample Title Element

(a) A title element with h1 tag

```
<h1>
  A Sample Title Element
</h1>
```

(b) Underlying source code of (a)

A Sample Title Element

(c) A title element with p tag

```
<p style="display: block;font-size: 2em;
margin: 0.67em 0.67em 0px 0px;
font-weight: bold;">A Sample Title Element
</p>
```

(d) Underlying source code of (c)

Figure 4.5: Two identical title elements with different HTML tags

In order to solve these problems, we propose a probabilistic approach to automatically identify the roles of visual elements. In our approach, we first build a modifiable knowledge base, which aims to describe semantic and structural attributes of visual elements comprehensively. Converting the knowledge to a set of rules, we calculate a likelihood score for each role for a visual element. Finally, we assign the role which has the highest likelihood score to the visual element.

The main process of this component can be divided into two modules: rule generation and role detection. In rule generation module, a comprehensive knowledge base is constructed and the knowledge is then converted to a set of rules in a certain syntax. Afterwards, role detection module applies these rules on visual elements. At the end of the process, a tree of labeled visual elements is created.

4.3.1 Rule Generation

Our knowledge construction work started with definition of roles of visual elements. Then, we analyzed a set of pages to determine characteristics of these roles in web pages. Finally, we formalized the role set and their characteristics in an ontological knowledge base.

We mainly took WafA as a basis and after a set of modifications, we compared it to WAI-ARIA and the role set of HTML5. Although WafA is a very rich knowledge resource, our initial experiments with WafA showed that it needs some modifications and extensions. First of all, node and collection concepts are irrelevant in our

study; therefore, we only consider atom and chunk concepts. Then, WAFa includes some role definitions, which are specific subgroups of general roles. The main problem with these roles is that, they have some very specific characteristics, so that; automating the process based on such definitions may have resulted in false positive role assignments. For example, *Link* role is divided into five sub-roles, which are *ReferentialLink*, *AssociativeLink*, *StructuralLink*, *SkipLink* and *ToTextOnlyPage*. These specific role definitions have a few or no distinguishing characteristics when we compare one another, so that, it is not possible to define rules to select a particular role. Consequently, all these sub-roles are included as *Link* in our ontology. However, if a role detection based application needs such detailed classification, it is possible to include these roles in the knowledge base. Finally, there are some missing roles in WAFa, especially for classification of textual and representational contents of the page, such as main content and complementary content.

Following roles are transferred from WAFa to our knowledge base:

- *Advertisement* is a graphical element which aims to direct users to an external page for commercial purposes. This role also covers *AdvertisementBanner* in WAFa.
- *BreadcrumbTrail* is a list of links and each link directs to a page in the hierarchical structure of the web site. A highlighted link refers to current page. This role is originated from *NavigationalBreadcrumbTrail*, *LocationBreadcrumb* and *AttributeBreadcrumb* and *PathBreadcrumb* roles in WAFa.
- *Caption* is a short text for a table or a figure. This role is originated from *FigureCaption* and *TableCaption* roles in WAFa.
- *Citation* is a quotation or a reference to an external source.
- *Copyright* is copyright note, usually located at the bottom of the page.
- *Figure* is a set of special graphics and caption objects.
- *Footer* is a container which is positioned at the bottom of the page. This role also covers *RunningFooter* in WAFa.
- *Header* is a container which is positioned at the top of the page. This role also covers *RunningHeader* in WAFa.
- *Icon* is a symbol for representing a tool or object in the page.
- *Label* is used with a form element to identify the form element.

- *Link* is used to navigate user to internal or external pages. This role is originated from *ReferentialLink*, *AssociativeLink*, *StructuralLink*, *SkipLink* and *TextOnlyPage* roles in WAFa.
- *List* is an array of items, such as links or content. This role is originated from *DefinitionList*, *NavigationalList*, *Bookmark*, *Favourites*, *ShoppingCart*, *Directory*, *SiteMap*, *WebDirectory*, *FAQ*, *HistoryList*, *Index*, *SiteIndex*, *TableOfContent*, *OrderedList*, *UnorderedList* and *ReferentialChunk* roles in WAFa.
- *Logo* is used to orient users to page itself, or external sources.
- *Menu* is a list of links for navigation purpose. This role is originated from *DropDownLinkMenu* and *LinkMenu* roles in WAFa.
- *MenuItem* usually used to navigate users to internal pages.
- *SearchEngine* is used for searching a content. It usually contains a text editor, a button and labels.
- *Sidebar* is a container which is positioned on the left or right side of the main content.
- *Separator* is a separator text or image between two distinct visual element. This role is originated from *SymbolSeparator*, *CommaSeparator*, *CopyrightSymbolSeparator*, *DashSeparator*, *TriangleLeftSeparator*, *TriangleRightSeparator* and *Vertical-BarSeparator* roles in WAFa.
- *SpecialGraphic* is an image which has a special meaning in the content.
- *Table* is a tabular array which consists of a set of rows and columns. This role is originated from *DataTable* and *LayoutTable* roles in WAFa.
- *Title* is an identification name, which is given to a section. This role also covers *Heading*, *Headline* and *SectionHeading* roles in WAFa.
- *TitleBanner* represents the page title or logo.
- *Toolbar* is a list of tool icons.

In order to extend the coverage of the roles in WAFa, it was also compared to WAI-ARIA. Our comparison showed that, both ontologies include similar roles; however, there were some noteworthy roles which appear in WAI-ARIA but are missing in WAFa. One major shortcoming in WAFa is that, there is no explicit definition for visual elements that contain textual content of the web page. These visual elements can be divided into two with respect to the relevancy of their content with the main

content of the web pages. First group contains overall or a specific part of the main content of the web pages. This content is unique to a particular page; therefore, it does not appear repeatedly in more than one page in a web site. The other group contains complementary content, which, unlike main content, appears in multiple web pages in a website. Finally, a role definition for describing form elements such as text areas, buttons and checkboxes was included in our knowledge base. As a result, following concepts were also taken from WAI-ARIA to extend the coverage of our knowledge base:

- *Article* contains a set of paragraphs, especially the main content of the page.
- *ComplementaryContent* is a content element which is not the main content of the element, rather a supplementary content.
- *Interaction* consists of form items such as input box, button and combo box.

HTML5 also provides a role attribute¹ based on WAI-ARIA. Since we already analyzed WAI-ARIA, we did not take any roles from HTML5 role set.

Finally, it was observed that, some visual elements do not have a specific role in the page, rather they are used to assure unity of other visual elements. To describe such kind of visual elements and overall body of a web page, we included following roles:

- *Body* represents the root element of the page.
- *Container* is a composite visual element which contains smaller sub elements.

During knowledge base generation process, we so far focused on determining the role set. Following work included analysis of visual elements in web pages and description of the characteristics of the roles.

We analyzed a set of web pages including simple and long-tail pages from different genres. These pages were selected among most popular pages listed in Alexa². After this analysis, it was observed that, each web element has a set of attributes which affect its representation in the page or its interaction with agents such as screen readers. It is possible to systematically characterize the roles of visual elements by using these attributes. In our investigations, it was noticed that the following properties

¹ <http://www.w3.org/TR/html5/dom.html#aria-role-attribute>, Last access: 20.07.2014

² <http://www.alexa.com/>, Last access: 20.07.2014

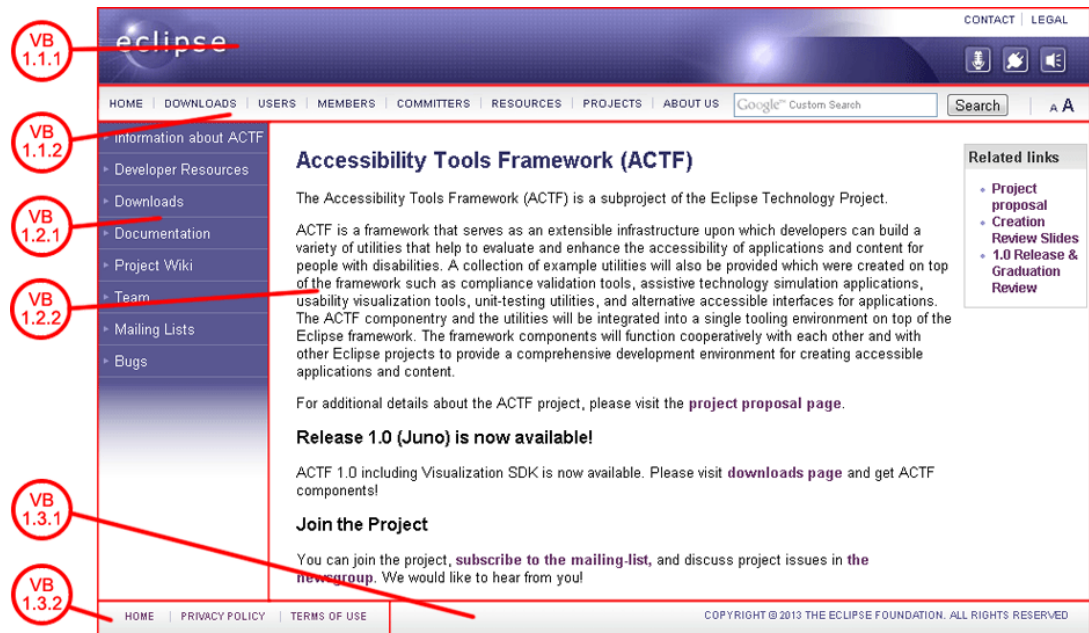


Figure 4.6: Visual elements in ACTF Home Page

affect how visual elements are used and represented:

- Underlying HTML/HTML5 tag;
- Children and parent visual elements in the underlying DOM tree;
- Size and position of the visual element in the page layout;
- Border and background color of the visual element;
- Some attributes including `onclick`, `for`, `onmouseover`, etc.;
- Visual style properties (font-size, color, etc.) of the visual element;
- Some specific keywords which appear in the textual content of the visual element;
- Some specific keywords which appear in the `id`, `class`, `role`, `src`, `background-image` attributes of the visual element.

Figure 4.6 illustrates the relation between some properties and corresponding roles of visual elements. Here, VB.1.1.1 represents the *Header* of the web page. Visual elements in *Header* role are generally placed at the top of the page and optionally contain a *Logo*, *Search Engine* or *Menu* child. Underlying HTML tag of a *Header* is either `header` or `div` in common; however, other HTML tags are rarely used to create a *Header*, such as `table` tag. Moreover, the visual element is associated with visual style definitions or client side events by its `id` or `class` attributes. For *Header*,

generally a common set of keywords such as "header" or "hdr" are used; however, there is no restriction for such keywords. Similar to the tag example, designer may assign an irrelevant `id` or `class` attribute value; however, "header" keyword in these attributes increases the probability of being a *Header*.

Regarding the properties which affect usage and representation of visual elements, we listed possible characteristics of the roles in our role list. During ontology creation, these characteristics were converted into formal representation to describe the knowledge about each role.

After identifying the roles of visual elements and their basic characteristics, we formalized them in an ontology. Our ontology described the heuristic roles in class level, where the instances of these classes are the visual elements in web pages. Moreover, our ontology included a set of object properties to define relations between classes and data types.

Class definitions in our ontology consist of basic main classes, which describe the attributes of visual elements. The main classes in the ontology are `tag`, `position`, `attribute` and `style` to describe HTML tags, specific positions of visual elements, some HTML based attributes and visual styles respectively. Furthermore, we defined sub-classes for describing these characteristics in detail. These subclasses are provided in Appendix C in OWL concrete abstract syntax. Moreover, our ontology included two main classes: `atom` and `chunk`. `atom` class is a base class for describing atomic units in the page; while `chunk` class describes composite visual elements. All of the roles are defined as subclasses of either `atom` or `chunk` classes with respect to their granularity.

The relations between classes and data types were defined on a set of object properties. Example 4.3.1 illustrates a sample object property definition. Here, domain of this object property is union of `atom` and `chunk` classes and its range is `position` class. It is mainly used to specify in which positions a visual element appears in the page layout. All of the object properties with their descriptions are given in Appendix D.

Example 4.3.1.

```
</owl:ObjectProperty>
```

```

<owl:ObjectProperty rdf:about="#in_position">
  <rdfs:comment><![CDATA[Specifies the positions which an atom or
    a chunk may have.]]></rdfs:comment>
  <rdfs:label>in_position</rdfs:label>
  <rdfs:factor>2</rdfs:factor>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#atom"/>
        <owl:Class rdf:about="#chunk"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range>
    <owl:Class rdf:about="#position"/>
  </rdfs:range>
</owl:ObjectProperty>

```

In order to improve the accuracy of the results, an iterative approach has been followed in object property specification. According to the initial experiments with our knowledge base, it has been noticed that some object properties gave more accurate information about the role of a visual element. As an example, the keywords in the `id` or `class` attribute of a visual element give more significant cue for its role than the size of visual element. A possible explanation for this is that, many visual elements with different roles in the web page may have the same size, while `id` and `class` attributes are generally unique to a particular role; therefore, they provide more descriptive information. Considering this, we defined an annotation factor value to differentiate the weights of the object properties. In this example, `has_id` object property had higher factor value than `has_size` object property.

Similarly, some object property values give more accurate information than other values for specific roles and object properties. For instance, an element may be a *Title* element, if it contains a node with either `p` tag or `h1` tag. However, the element with `h1` tag is more likely to be a *Title* element, than those having a `p` tag. To handle such cases, three levels of object properties, such as `must_have_tag`, `has_tag` and `may_have_tag` were defined where `must_have_tag` had the highest factor annotation value and `may_have_tag` had the lowest factor annotation value. Our manual evaluation showed that, these enhancements improved the accuracy of the role assignments.

Finally, all of the roles and characteristics were formalized in the ontology by using

main classes and object properties which were previously defined. Based on the granularity of each role, it was represented as a sub-class of either `atom` class or `chunk` class. A sample part of role definition for *Header* elements is given in Example 4.3.2. In this example, the role definition and some characteristics of this role are illustrated. According to this definition, a *Header* is likely to have a `header` or `div` tag. Also, its `id` and `class` attributes are likely to contain one of "header", "hdr", "hd" or "heading" keywords.

Example 4.3.2.

```
<owl:Class rdf:about="#Header">
  <rdfs:comment><![CDATA[Is typically printed at the top of a page
    and includes, for example, a company logo, the title of the page,
    list of links and sometimes a search engine.]]></rdfs:comment>
  <rdfs:label>Header</rdfs:label>
  <rdfs:subClassOf rdf:resource="#chunk"/>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#chunk"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#must_have_tag"/>
      <owl:hasValue rdf:resource="#Header"/>
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#has_tag"/>
      <owl:hasValue rdf:resource="#Div"/>
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#has_id"/>
      <owl:allValuesFrom>
        <owl:Class>
          <owl:oneOf rdf:parseType="Collection">
            <owl:Thing
              rdf:about="http://www.w3.org/2001/XMLSchema#header"/>
            <owl:Thing
              rdf:about="http://www.w3.org/2001/XMLSchema#hdr"/>
            <owl:Thing
              rdf:about="http://www.w3.org/2001/XMLSchema#hd"/>
            <owl:Thing
              rdf:about="http://www.w3.org/2001/XMLSchema#heading"/>
          </owl:oneOf>
        </owl:Class>
      </owl:allValuesFrom>
    </owl:Restriction>
    ...
  </owl:intersectionOf>
</owl:Class>
```

Having defined the roles and their characteristics in the ontology, our knowledge representation work was completed. During the following process, we focused on using this knowledge to automatically detect the heuristic roles of visual elements.

After the knowledge was formally represented in an ontology, then this ontology was

processed to construct heuristic rules in proper format to apply on visual elements which were identified in Section 4.2. The rules mainly check related attributes of a visual element and increment the likelihood score of corresponding role for it.

First of all, a data type was defined to interpret visual element with its attributes in the rules. This data type was named as `block` and included a set of slots which kept the corresponding attributes for each object property. Example 4.3.3 presents the definition of `block` data type.

Example 4.3.3.

```
(deftemplate block
  (multislot is_composite)
  (multislot has_size)
  (multislot has_order)
  (multislot is_atomic)
  (multislot has_keyword)
  (multislot in_position)
  (multislot font_size)
  (multislot border)
  (multislot font_color)
  (multislot has_style)
  (multislot contains_linebreak)
  (multislot has_background)
  (multislot word_count)
  (multislot has_child)
  (multislot has_sibling)
  (multislot has_parent)
  (multislot has_type)
  (multislot has_id)
  (multislot must_have_tag)
  (multislot relative_size)
  (multislot doc)
  (multislot has_tag)
  (multislot has_list_style)
  (multislot has_attribute)
  (multislot font_weight)
)
```

To construct the rules, we parsed the ontology and converted each restriction definition of the role classes to rules in proper format in runtime. However, this approach had the disadvantage of long response time and waste use of resources. In order to prevent such drawbacks, we developed a module to check whether the ontology is modified. This module works as following: First, it calculates the MD5 checksum of the latest version of the ontology file and store this value in a local file. Before each execution, MD5 checksum of the ontology in host is calculated and compared to the value stored in the local file. If these two values are not equal, it means that ontology was modified and the rules should be regenerated. In such cases, system constructs

a local file, which has a special file format for rule definitions. On the other hand, in case of equality, meaning that the ontology remained the same as the previous run, the local file, which was previously saved, is loaded to working memory of rule engine. With this enhancement, time required to parse the ontology and convert knowledge to the rules was reduced.

Example 4.3.4 illustrates a sample rule definition for the *Header* role:

Example 4.3.4.

```
(defrule Header00
  (block (must_have_tag $? /*header.*/* $?))
  => (bind ?*Header* (+ "8" ?*Header*)))
```

Here, the initial part after rule name denotes which data type the rule applies and in which condition it is fired. In this example, rule is applied on `block` objects and it is fired if visual element has a `header` tag. Second part, which comes after `=>` operator, denotes the action if the visual element satisfies the required condition in the first part. In this example, the likelihood score for *Header* is incremented by 8, which is the value of factor annotation for this rule.

4.3.2 Role Detection

For identifying the roles of visual elements, a rule engine is used. The benefit of using a rule engine is that it improves the performance of the system in both ontology parsing and reasoning. First of all, the rule definitions, their object properties and restrictions are converted to formal rules, so that, the system does not parse the ontology in each execution. Therefore, the time required to parse the ontology is reduced. Then, querying on the ontology to retrieve information is a time consuming task. However, it is more efficient to assert a visual element to rule engine and fire the rules on it. Our initial experiments with rule engine also resulted in a significant performance gain in our system.

Our ontology defines the roles of visual elements in class level. The individuals, which are the instances of these classes, correspond to visual elements in web pages. Before applying heuristic rules on visual elements, they are converted to facts. Each

fact includes a set of visual or structural attributes of corresponding visual element. Unlike rules and role definitions, these facts are created in each execution based on the input web page and its visual elements.

Following process is applied during heuristic role detection of visual elements: First, an instance of rule engine is constructed in the system. A template to define the structure of facts corresponding to visual elements and initial likelihood scores, which are globally set to 0, are loaded to the rule engine with an external file. This file also includes the rule set to be applied on the facts to reason. As was mentioned earlier, these rules are retrieved from the relations between classes and data types in the ontology. Then, for each visual element, a clone of the rule engine is created. Cloning the initial rule engine at each iteration has the advantage of reducing the time required to initialize the working memory for each visual element. After the fact, which represents a visual element, is asserted to the working memory of the rule engine and reasoning process is run, the rules in the working memory fire on the fact. If the fact satisfies the condition in the rule, the global variable of the heuristic role of corresponding rule is incremented as defined in the action part of the rule. Likelihood score of a role is calculated with the Equation 4.1:

$$S_i = \sum_p \sum_n a_p * p(attr_n) \quad (4.1)$$

Here, the likelihood score of a role i is represented as S_i . p represents the object properties of the role i , which return either 1 or 0 with respect to whether the visual element satisfies the requirement of the corresponding object property with one of its related attributes. m is the total number of attributes of the visual element related with the object property p . a_p is the factor annotation of p , which denotes the weight of the object property. Finally, $attr_n$ is an attribute of visual element which is checked in the condition part of the rule which corresponds to p .

After all rules are fired and the likelihood scores are calculated for each role, the role with the highest likelihood value is assigned to the visual element. The highest likelihood score is calculated with Equation 4.2:

Block	Path	Block	Role
VB.1	/HTML/BODY	VB.1	Body
VB.1.1	/HTML/BODY/COMPOSITE	VB.1.1	Container,
VB.1.1.1	/HTML/BODY/DIV	VB.1.1.1	Header,
VB.1.1.2	/HTML/BODY/DIV[2]	VB.1.1.2	LinkMenu,
VB.1.2	/HTML/BODY/COMPOSITE	VB.1.2	Container,
VB.1.2.1	/HTML/BODY/DIV[3]	VB.1.2.1	LinkMenu,
VB.1.2.2	/HTML/BODY/DIV[4]	VB.1.2.2	Container,
VB.1.3	/HTML/BODY/DIV[5]	VB.1.3	Footer,
VB.1.3.1	/HTML/BODY/DIV[5]/UL	VB.1.3.1	LinkMenu,
VB.1.3.2	/HTML/BODY/DIV[5]/P	VB.1.3.2	Copyright,

(a) Tree of visual elements

(b) Roles of visual elements

Figure 4.7: Overall results for ACTF Home Page

$$R = \max(attr_1, attr_2, \dots, attr_m) \quad (4.2)$$

The result of this process indicates the heuristic role which the visual element is more likely to have with respect to its visual and structural attributes. Assertion and firing procedure repeats for each visual element in the page. After the role of a visual element is detected in each iteration, the tree of visual elements is updated. At the end of overall process, our tree of unlabeled visual elements turns to the tree of labeled visual elements.

Figure 4.7 illustrates the output of the system after execution of each component. The visual element structure of the ACTF Home Page after visual element identification is represented in Figure 4.7a. Also, Figure 4.7b shows assigned roles of each visual element after role detection process. In this case, *Header* and *Link Menu* roles are assigned for VB.1.1.1 and VB.1.1.2 respectively based on the attributes of these visual elements.

4.4 Implementation

We implemented our system on Accessibility Tools Framework (ACTF) on Java platform³. ACTF is a framework and extensible infrastructure, which enables developers

³ <http://www.eclipse.org/actf/>, Last access: 20.07.2014

to build a variety of utilities for improving the accessibility of applications and content for people with disabilities. ACTF provides a IE based browser and interfaces to control this browser, such as changing textual content of a visual element. Using these interfaces, DOM structure of a web page and its visual rendering properties can be retrieved and manipulated.

In rule generation, we also used some external tools and libraries. First of all, the ontology is parsed with OWL API. Moreover, we used HermiT reasoner to perform basic querying, such as obtaining all subclasses of the base classes and their characteristics defined by object properties. Finally, Jess, the Rule Engine for the *JavaTM* Platform is used to reason on the predefined rules.

We also committed our implementation to ACTF and it is available under the terms of the Eclipse Public License v1.0⁴.

4.5 Summary

This chapter has described our approach to identify visual elements in web pages and overall system architecture. The system has two main components, which are visual element identification and automatic role detection.

The visual element identification component is responsible for identifying visual elements in a web page. This component is based on the VIPS algorithm. However, VIPS has some limitations, which we categorized as corrective and adaptive limitations. During our implementation, we performed some improvements on the original algorithm. The component returns a tree of visual elements based on structural and representational features of them.

The next component, automatic role detector, receives the tree of visual elements and detects the corresponding roles of them. It is based on a probabilistic approach, which takes a full description of a visual element and returns the role which is more likely with respect to its attributes. After having determined the role set and described their characteristics, this knowledge was formally represented in an ontology. Reasoning

⁴ <http://www.eclipse.org/legal/epl-v10.html>, Last access: 20.07.2014

to determine corresponding role was performed on a rule engine by using a set of rules that originated from this ontology. This component constructs the output of the overall system, which consists of a tree of visual elements with corresponding roles assigned.

When the role set and visual element attributes which describe these roles are analyzed, it can be seen that proposed approach extends previous methods with its comprehensive knowledge base. Therefore, this approach can be used in various fields, such as web page adaptation for web accessibility or information retrieval. As a case study, we implemented a transcoding architecture based on our approach. Chapter 5 explains the implementation process of transcoding application.

In addition, to validate our proposed approach, we have conducted both technical and user evaluations. Chapter 6 presents the evaluation procedure and the results.

CHAPTER 5

WEB PAGE TRANSCODING

Transcoding is one research field in which both web page segmentation and role detection of visual elements are used in general. To demonstrate a use case for our visual element identification and role detection approach, a transcoding application has been developed. Unlike previous transcoding methods, our approach is based on eye tracking data to better adapt user navigation behaviour on web pages. In this chapter, we present a case study design and implementation for web page transcoding based on eye tracking data.

This chapter is organized as follows: Section 5.1 explains basic transcoding strategies which benefit the aspects of visual element identification and role detection. Then, Section 5.2 describes eye tracking data and how it is processed for our transcoding approach. Section 5.3 follows by providing an overview of overall system architecture. Finally, implementation details of transcoding application are presented in Section 5.4.

5.1 Transcoding Strategies

In this section, we discuss basic transcoding strategies and how they can benefit automatic identification of visual elements and their roles. These strategies mainly aim to address some basic web accessibility problems when web pages are accessed by using either screen readers, mobile devices or both. In this study, we only focus on the methods which different aspects of visual element identification and role detection can be utilized. The other approaches that include layout modification or representa-

tional manipulation are not covered in this study.

5.1.1 Selective Representation

Web pages include many visual elements and some of these elements may not be interesting to some users. Representation of overall web page causes some problems in both screen readers and mobile devices. First of all, screen readers typically follows the DOM order to read all visual elements. This increases the access time to a particular visual element which the users are interested in. Moreover, mobile devices have limited resources to display overall web page, such as bandwidth and memory. Finally, they produce complex representation in the small sized screens.

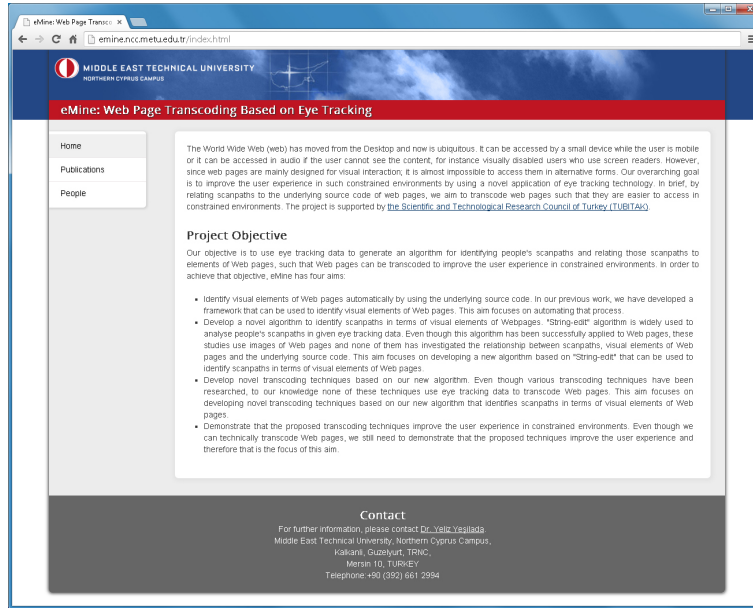
Visual element identification and role detection can help to solve these problems by selecting a specific visual element at each time and representing it on the screen. Identification of visual elements helps to interpret web page in a simpler structure and select only a specific part of the web page. Moreover, selection can be performed according to the roles of the visual elements, so that, only specific elements that correspond to a particular role can be selected for representation.

This method simplifies the page content; provides simplistic view in small screen devices and reduces the time to reach important content. Figure 5.1 illustrates the effect of this operation in a web page. Web page in Figure 5.1a is represented by selecting only the main content and removing remaining visual elements as in Figure 5.1b.

5.1.2 Summarization

There are many approaches which provide a summary of web pages into form of a table of contents [3, 58] or as thumbnail view [123, 33, 32, 125, 119, 8]. Both approaches have similar aspects; however, a thumbnail representation may cause accessibility issues for disabled users in particular.

A typical table of content represents only the titles or headings in a content. Therefore, to display a summary of a web page, first the titles and headings must be detected. To achieve this, visual elements in overall page must be identified. Among



(a) Desktop view



(b) Mobile view

Figure 5.1: eMINE - Home page in both desktop and mobile

all visual elements, selecting only titles and headings requires a role detection process. Alternatively, a web page can be summarized with respect to the roles of visual elements. In this approach, the table of content presents the role of each visual element instead of a title to express the content. Similarly, representing a web page with thumbnails requires visual element identification and role detection, since each different type of visual elements should be represented with a specific thumbnail.

5.1.3 Sequential and Direct Access

Similar to pagination mechanisms in web pages, visual elements can be placed in a path and be accessed sequentially. In the literature, a similar approach [107] was used to order visual elements based on their importance and these elements were accessed in a particular order. Direct access, on the other hand, gives user the ability to jump to a specific visual element after users select a certain item in the path. However, putting visual elements in an order and accessing a particular visual element directly primarily require identification of overall visual elements in web page.

A typical path can be represented in terms of visual elements or underlying roles. In

the former case, the path consists of visual element names or XPATHs to associate each item in the path with corresponding visual element. In this case, sequential or direct access immediately returns the corresponding visual element. If the path includes a role set, then a set of visual elements correspond to each item in the path, since there may be more than one visual element with a specific role in a web page.

In order to sum up, visual element identification and role detection processes can effectively be used in web page transcoding. The main transcoding strategies are selective representation, summarization, and sequential and direct access. All three techniques can benefit different aspects of both visual element identification and role detection. In this study, we aim to transcode a web page based on eye tracking data. The eye tracking data provides significant information about in which order a user looked at a particular visual element [126, 43]. In our transcoding approach, we provide sequential access to the visual elements in the path and selective representation with respect to the position of user in this path. Moreover, we summarize the web page and enable direct access to the visual elements in the summary. The next section explains the path construction process in terms of both visual elements and roles.

5.2 Eye Tracking Based Transcoding

In our transcoding approach, we use eye tracking data to transform web pages. The eye tracking data used in this study is retrieved from the preliminary results of eMine Scanpath Analysis Algorithm [126, 43]. This algorithm uses our visual element identification implementation and generates a common scanpath by using eye tracking data and a tree of visual elements of a segmented web page. In the context of that study, a set of user experiments were conducted and eye tracking data of the participants for six pages was collected. To generate the scanpath, the algorithm first projects the fixations of eye tracking data over the visual elements in web pages, and with respect to order of viewing, it generates a path for each participant. Then, among a set of scanpaths, the most common scanpath is calculated for a particular web page in the following way: First of all, two most common scanpaths are detected by using Levenshtein Distance. Then, the common scanpath of these two scanpaths are found with Longest Common Subsequence method. The new scanpath is inserted into the

scanpath set, and two scanpaths are removed. This process iterates until only one scanpath is left in the list. At the end, a common scanpath of visual elements for a web page is generated.

In the context of user experiments of eMine Scanpath Analysis Algorithm, six pages were used with different visual complexities. These pages were Apple.com and Babylon.com with low visual complexity; Avg.com and Yahoo.com with medium complexity, and Godaddy.com and Bbc.co.uk with high complexity. In the experiments, users were told to complete a set of specific searching and browsing tasks. While participants were completing these tasks, eye tracking data of each participant was recorded to construct common scanpaths for each of the web pages. In our transcoding work, we use these six pages and their corresponding common scanpaths.

Figure 5.2 represents Babylon Home Page with the visual elements in common scanpath are highlighted. In this web page, common scanpath includes visual elements indicated by ①, ②, ③, ④ and ⑤. This example can be interpreted as that participants in general looked at the visual element which is indicated by ① and then followed through ②, ③, ④ and ⑤ [42]. Similarly, eMine Scanpath Analysis Algorithm calculated a common scanpath for all six pages. These common scanpaths consist of a list of visual elements, in which each visual element is interpreted with its name. Example 5.2.1 illustrates a common scanpath with five visual elements in JavaScript Object Notation¹ (JSON) format. The other common scanpaths are illustrated in Appendix E.

Example 5.2.1.

```
{
  "main_path": [
    "VB.1.1",
    "VB.1.2.1",
    "VB.1.2.2",
    "VB.1.2.3",
    "VB.1.2.4"]
}
```

We processed these common scanpaths to build a common role path. Our intent in this process was to define a general role path to use in transcoding mechanism. Therefore, we detected the role of each visual element in the scanpaths and converted visual element names to underlying roles. We refer these new paths as role paths.

¹ <http://json.org/>, Last access: 09/07/2014



Figure 5.2: Babylon Home Page

Table 5.1 represents the role paths of each web page for both searching and browsing task results.

The main purpose of this analysis was to generate a common role path by using the results of six pages. However, when we analyzed the role assignments and their relation with each other, we saw that, there were some problems with them. First of all, the role diversity was very narrow and role paths only contained *SpecialGraphic*, *Article*, *Title*, *Link*, *List* and *Container*. Moreover, some role paths contained only a single role. Similarly, many of the role paths contained duplicate role and in some cases, all roles in role path consisted of a single role. Another problem with these role paths was the high frequency of the *Container*. Although there were many instances

Table 5.1: Role paths of six web pages

Task	Page	Role Path
Searching	Apple	Container(<i>SpecialGraphic</i>)
	Babylon	<i>SpecialGraphic</i> → <i>SpecialGraphic</i> → <i>SpecialGraphic</i> → <i>SpecialGraphic</i> → <i>SpecialGraphic</i> → <i>SpecialGraphic</i>
	AVG	<i>Article</i> → Container(<i>SpecialGraphic</i> , <i>Title</i> , <i>Link</i>) → <i>Article</i> → Container(<i>SpecialGraphic</i> , <i>Title</i> , <i>Link</i>)
	Yahoo	Container
	Godaddy	Container(<i>Title</i> , <i>Article</i> , <i>Link</i>) → Container(<i>Title</i> , <i>Link</i>) → Container(<i>Title</i> , <i>Link</i>)
	BBC	Container(<i>Title</i> , <i>Link</i> , <i>List</i>)
Browsing	Apple	<i>SpecialGraphic</i> → <i>SpecialGraphic</i>
	Babylon	<i>SpecialGraphic</i>
	AVG	<i>Article</i> → <i>Article</i>
	Yahoo	Container → Container → Container
	Godaddy	Container(<i>Title</i> , <i>Article</i> , <i>Link</i>) → Container(<i>Title</i> , <i>Article</i> , <i>Link</i>) → Container(<i>Icon</i>) → Container(<i>Title</i> , <i>Article</i> , <i>Link</i>)
	BBC	Container(<i>Link</i> , <i>List</i>) → Container(<i>Link</i> , <i>List</i>)

of *Container*, the inner visual element roles which *Container* elements contain, differed from each other. Therefore, each *Container* which contained different types of visual elements should have been separated than others. For example, Container (*Title*, *Article*, *Link*) and Container (*SpecialGraphic*, *Title*, *Link*) should have been distinguished. These limitations showed that, generating a common role path to apply many different web pages required more than six pages to analyze.

In order to use in our transcoding use case and prepare a basis for future common scanpath studies on eye tracking data, we created an example role path. Example 5.2.2 illustrates the format of common role path in JSON format, similar to common scanpaths.

Example 5.2.2.

```
{
  "main_path": [
    role-1,
    role-2,
    role-3,
    ...,
    role-n]
}
```

This data structure simply defines a path for roles. Alternative way of thinking about web pages, is to consider them as a composite structure which includes a set of visual elements and these visual elements are the instances of a specific role. Our experiments with roles of visual elements have shown that, a role may have multiple

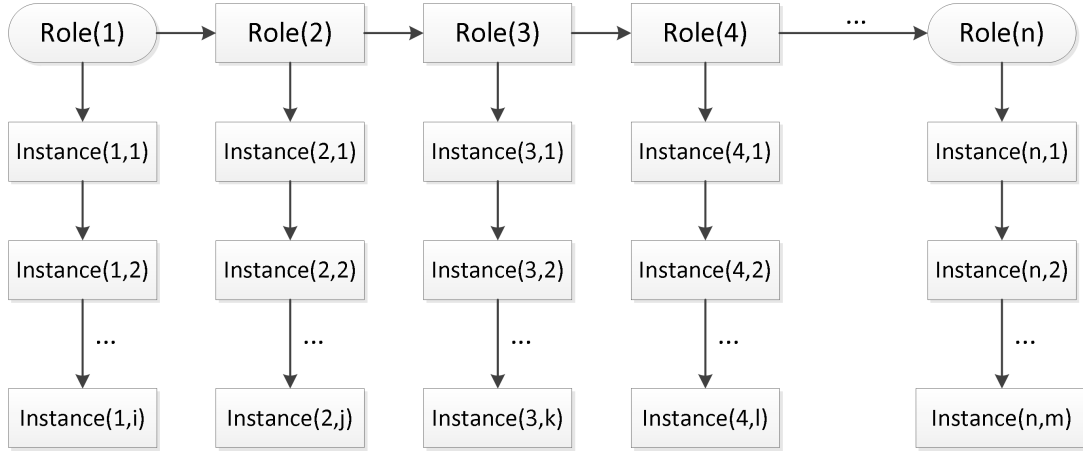


Figure 5.3: Projection over visual elements

instances in a web page. Therefore, we need to associate a set of visual elements with a role, rather than one-to-one matching. Figure 5.3 illustrates this association of each role in the role path with corresponding instances.

In order to sum up, our transcoding application is based on eye tracking data and this data is retrieved from the study of eMine Scanpath Analysis Algorithm as common scanpaths. In this study, we used two different set of paths, which are common scanpaths and common role paths. Common scanpaths shows the visual elements which are commonly viewed by the users in their viewing order. This data can be used to provide a transcoding mechanism for specific web pages. Moreover, we intended to construct a common role path by using common scanpaths and corresponding roles of visual elements in them. Our results showed that, this generalization procedure requires more extensive data set. However, to demonstrate the use of roles in transcoding, we proposed an example role path. The next section describes the use of these paths in transcoding and its overall system architecture.

5.3 System Architecture

In this section, we introduce our transcoding approach. The overall system architecture of our approach is shown in Figure 5.4. In brief, the system architecture consists of three main components, which are visual element identifier (indicated by ①), knowledge base for common scanpaths (indicated by ②) and visual element tree

cache (indicated by ③).

When a user sends a request to retrieve a web page, the visual elements in the web page are identified and corresponding roles are detected. Then, by using common scanpath or common role path, the visual elements or the roles are ordered in an array. Initially, only the first visual element in the path is displayed. After a certain interaction event, the state changes and the screen is updated with the next visual element in the path is shown.

The interaction events are designated to illustrate all transcoding strategies presented in Section 5.1. These events include sequential access to the visual elements in common scanpath or roles in common role path and their corresponding visual elements; direct access to certain visual elements and displaying a summary of overall web page in terms of visual elements or roles, as well as summary of a role in terms of visual elements.

In order to overwhelm the accessibility problems of both screen readers and mobile devices, we suggest a proxy based approach for web page transcoding. Such an approach appears to be the most convenient way for our application for a set of reasons. First of all, it simplifies the content to display interesting part of the web page based on the common scanpath or common role path. This reduces the time required to access these visual elements in screen readers. Moreover, visual element identification is handled in a proxy-server. Therefore, bandwidth and memory usage of mobile devices are also reduced. To simulate this architecture, we developed a demonstration of our transcoding approach. For experimental purposes, both proxy-server and client architectures were implemented in a single application. The next section describes the implementation process and different aspects of transcoding based on eye tracking data.

5.4 Implementation of Transcoding Application

As in visual element identification and role detection components, we developed our transcoding demonstration on Accessibility Tools Framework (ACTF) on Java platform. In the transcoding application, there are two modes: block based transcoding

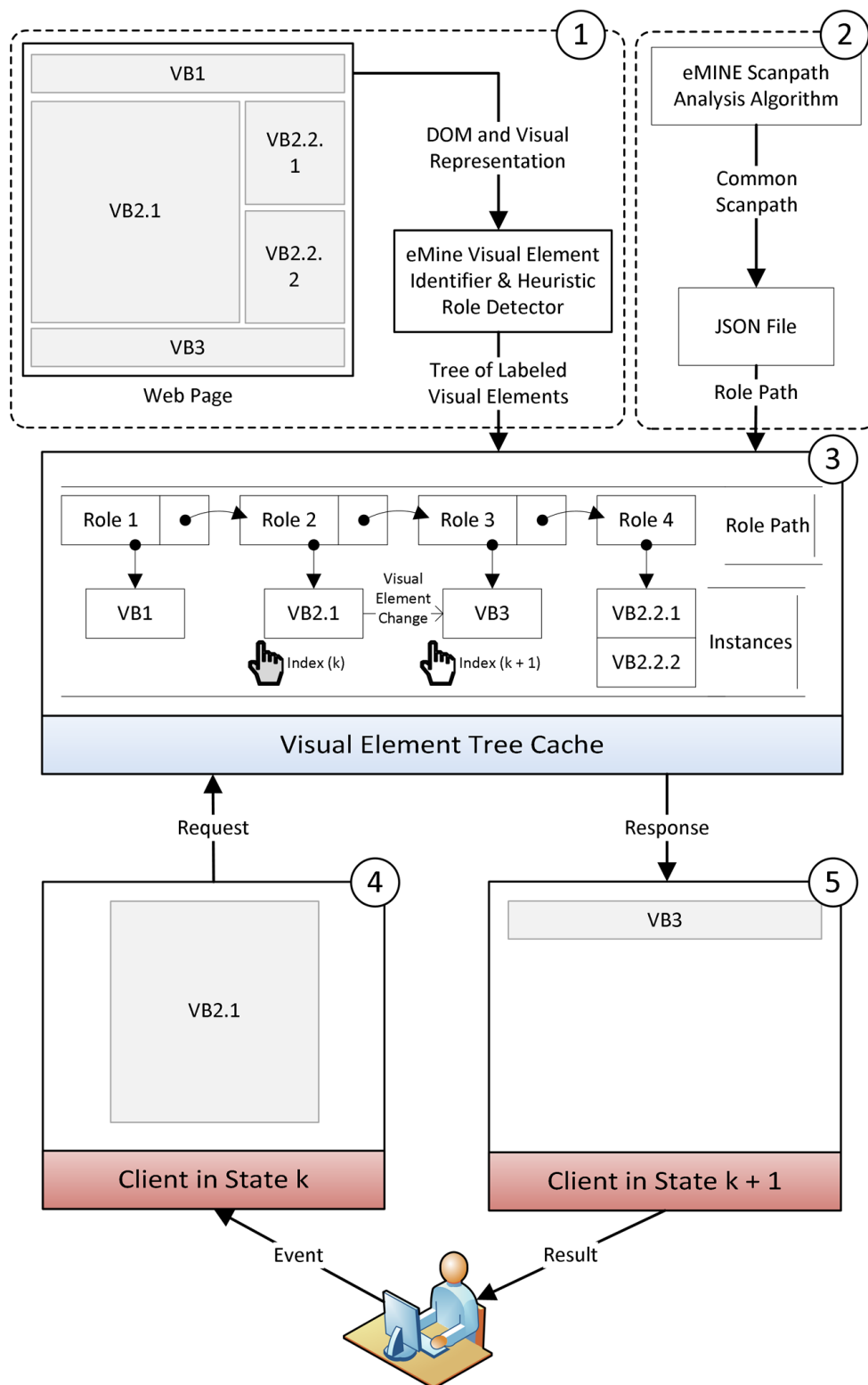
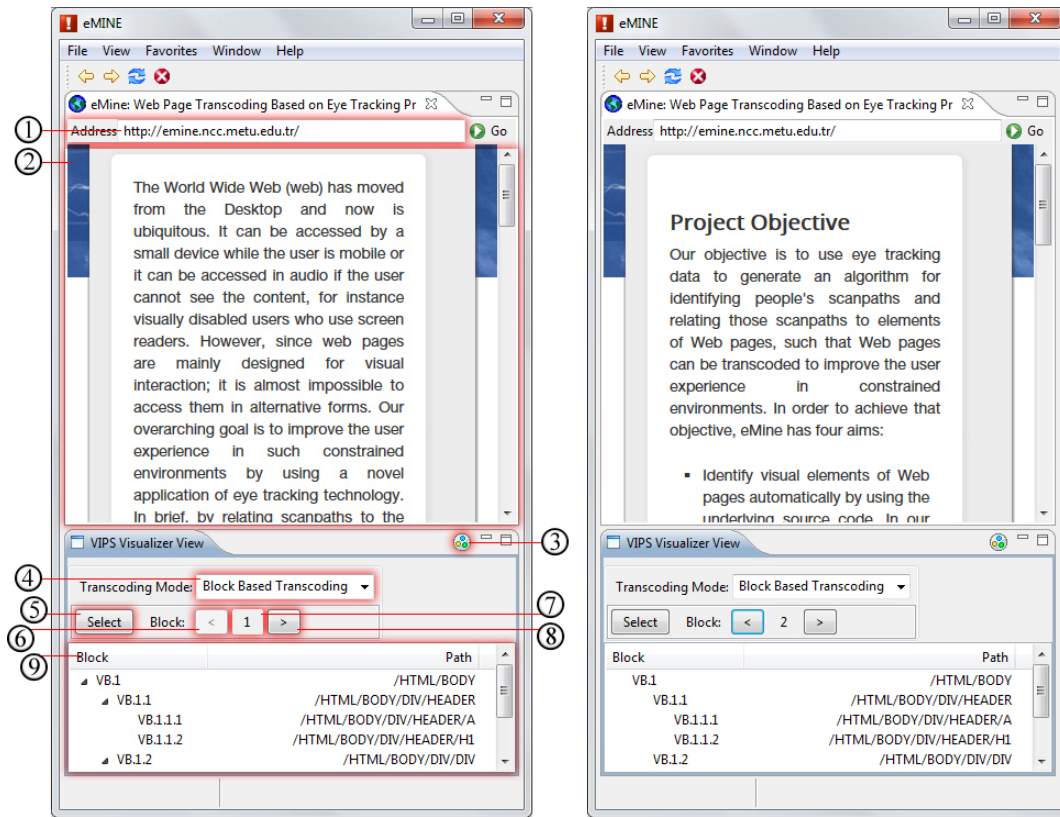


Figure 5.4: Proxy server application architecture



(a) 1st Block in Common Scanpath

(b) 2nd Block in Common Scanpath

Figure 5.5: eMINE - Home page in block based transcoding application

and role based transcoding. In block based transcoding, visual elements are ordered with respect to common scanpath of the particular web page. This common scanpath is loaded as a JSON file. In role based transcoding, visual elements in a web page are associated with their corresponding roles in the common role path. This role path is a generalized version of multiple common scanpaths and intended to be applied on all web pages.

5.4.1 Block Based Transcoding

In block based transcoding, a web page is transcoded based on its common scanpath. The common scanpath is loaded as a JSON file to the application. We use common scanpaths which were generated in the context of eMine Scanpath Analysis. The common scanpaths include a list of visual element names to associate each item in the path with corresponding visual element.

Figure 5.5a represents the demo application of block based transcoding approach. After user types the URL of a web page in the address bar (indicated by ①) and clicks on "Go" button, the content of the web page is loaded in its original form in the content panel (indicated by ②). When user clicks on segmentation button (indicated by ③), visual element identifier component segments the web page. This action also turns file selection button (indicated by ⑤) to enabled. With this button, user selects a JSON file which contains corresponding scanpath of the web page. After common scanpath is loaded into the application, only a specific visual element is displayed based on the items in common scanpath. Then, user can navigate on the common scanpath back and forth, by using previous visual element button (indicated by ⑥) and next visual element button (indicated by ⑧) respectively. The current index is also displayed (indicated by ⑦). In addition, hierarchical structure of visual elements of the web page is represented in visual element structure panel (indicated by ⑨).

Figure 5.5 also illustrates navigation between visual elements. Figure 5.5a shows the initial state of the application, in which the first visual element in the common scanpath is represented. After user clicks on next visual element button, current index is changed to point to the second item in the common scanpath. After this action, the display of the application updated, so that, the first visual element in common scanpath is removed and only the second visual element in the path is displayed. User can navigate through all visual elements in the common scanpath by using navigation buttons.

5.4.2 Role Based Transcoding

The other mode of our simulation application is role based transcoding. As mentioned earlier, we used an example role path. This role path was defined as in Example 5.4.1.

Example 5.4.1.

```
{
  "main_path": [
    "Article",
    "Title",
    "Footer"]
}
```

In this mode, first the roles of the visual elements in web page are detected and then

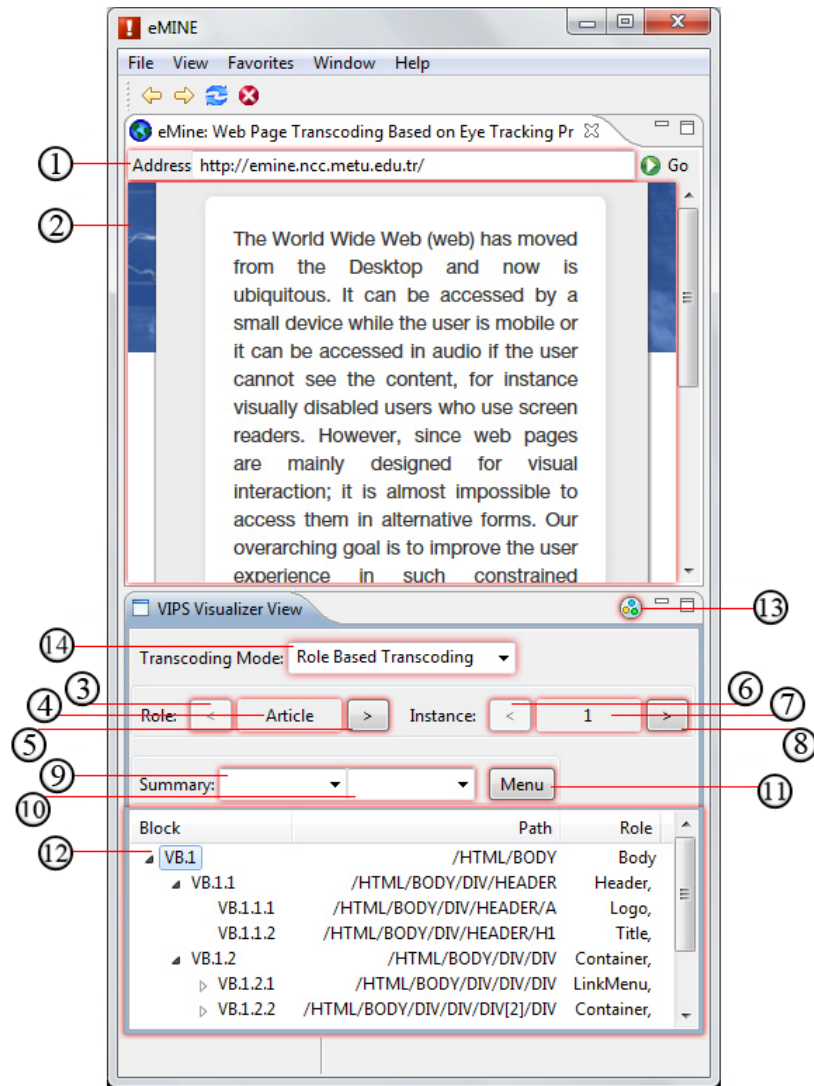


Figure 5.6: Role Based Transcoding Demo Application

visual elements are grouped with respect to their roles. Using the roles in the common role path, these visual elements are accessed directly or sequentially. Unlike block based transcoding, users do not load a JSON file in this mode. Instead, the JSON file is stored in a specific folder in the local file system. The transcoding application parses this JSON file in each execution of the transcoding algorithm. Since the knowledge base in the role detection algorithm was intended to be modifiable, the role path is also modifiable. Therefore, comprehensive role paths in future studies can be properly used with this approach.

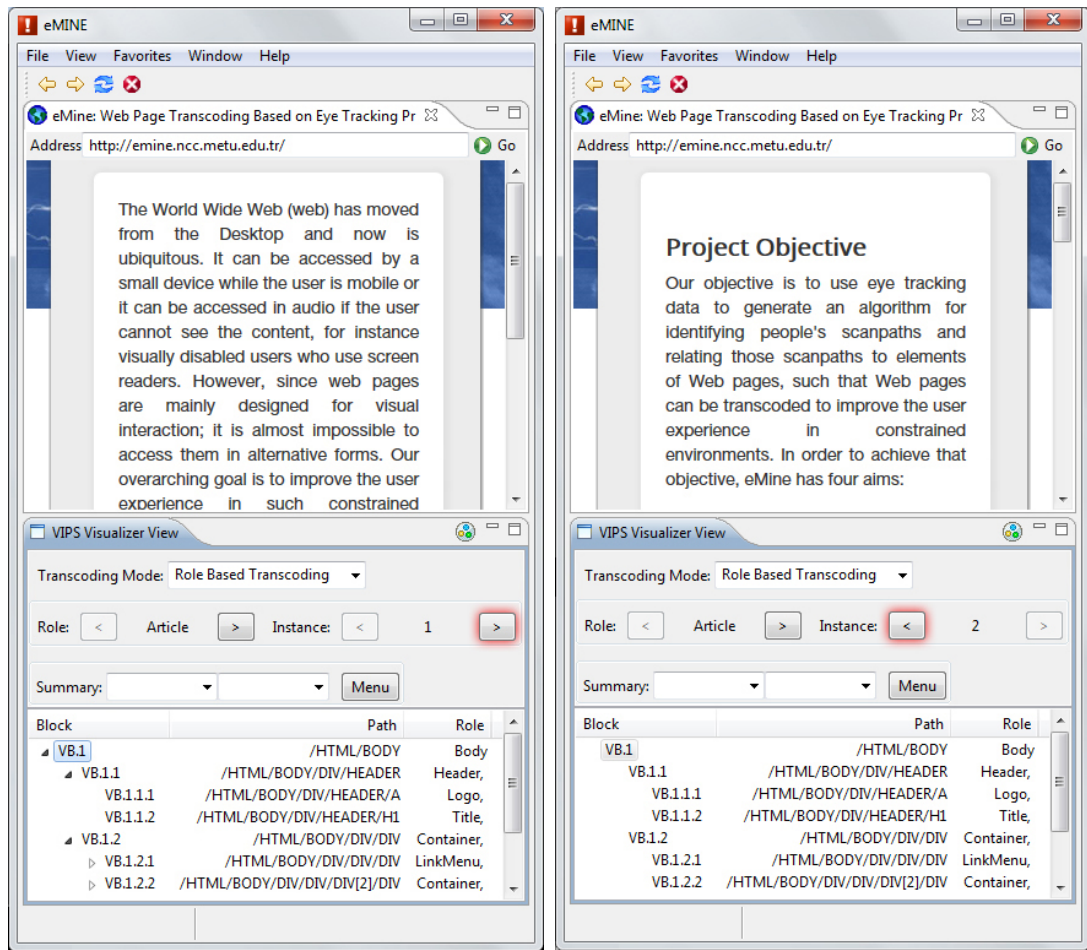
Figure 5.6 represents the demo application of role based transcoding approach. Similar to block based transcoding mode, it includes an address bar, a content panel, visual

element structure panel and a set of interaction items to navigate through visual elements in web page. After user clicks on transcoding button (indicated by ⑬), the application first segments the web page and detects the heuristic roles of the visual elements. Based on their roles, visual elements are grouped. At the initial state, the first instance of the first role in role path is displayed. Then, user can navigate through roles by using next role button (indicated by ⑤) and previous role button (indicated by ③). Also, user can use next instance button (indicated by ⑧) and previous instance button (indicated by ⑥) to travel through the instances of a role. However, the common role path may not include all the roles which appear in the web page. In order to solve this problem, a summary of the in terms of roles is provided (indicated by ⑨), as well as a summary of the current role (indicated by ⑩). Finally, by using menu button (indicated by ⑪), users can jump directly to the menu or menu item visual elements of the web page.

Figure 5.7 illustrates navigation between visual elements. In Figure 5.7a, user is in the initial state and application displays the first instance of *Article* role. There are two *Article* instances in the page; therefore, user may jump to the next *Article* instance. After user clicks on the next instance button, current instance of *Article* role is removed and the next instance of *Article* role in the page is displayed.

This transcoding method also provides direct access to the users. It is achieved by providing both a summary of the page structure in terms of roles and instances, and direct links to specific parts of the page. Figure 5.8 represents the summary functionalities of the demo application. As highlighted in Figure 5.8a, the list of the roles which appear in the web page is provided as a select box. If user select a role in the list, for example *Footer*, the current visual element is removed and only the first instance of corresponding role is displayed. Similar to the role summary, the summary of instances of the current role is also accessible. Figure 5.8b represents the summary of instances of *Article* role. If the user selects one instance in the list, current instance is removed and only selected instance is displayed.

Another type of direct access is handled by providing specific buttons for some roles. In Figure 5.9, the button which enables users to access menu visual element is represented. After user clicks on this button, the first instance of *Menu* or *Menu Item* is



(a) 1st instance of article role

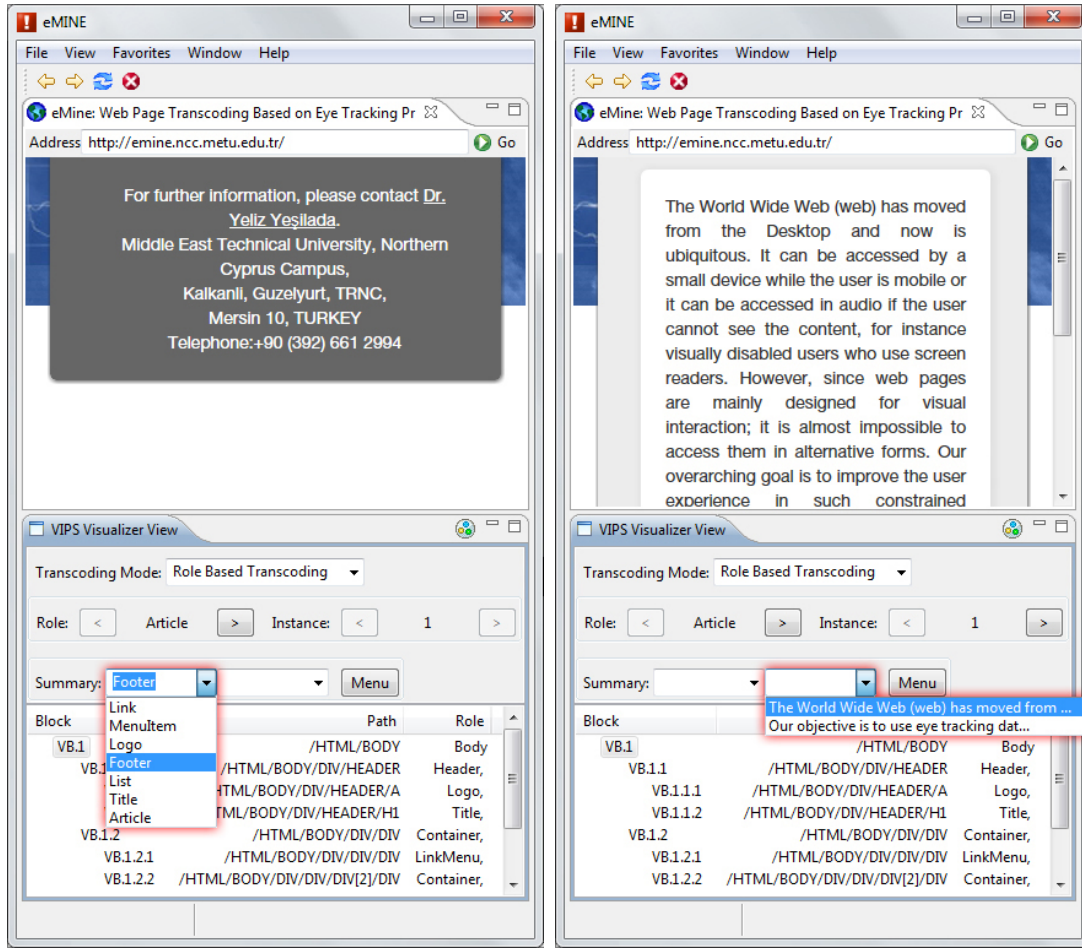
(b) 2nd instance of article role

Figure 5.7: eMINE - Home page in role based transcoding demo application

displayed. It is possible to extend functionality of direct access with respect to various needs of the users.

5.5 Summary

This chapter described our case study of web page transcoding based on eye tracking. Using our visual element identification and role detection approach, a system architecture was proposed and then simulation of the proposed approach was implemented. In this simulation, three different transcoding strategies were used, which are selective representation, summarization, and sequential and direct access. These strategies benefit the aspects of visual element identification and role detection.



(a) Role summary

(b) Instance summary

Figure 5.8: eMINE - Home page in role based transcoding demo application

In this chapter, the eye tracking data description was also provided. In our study, we used the results of eMine Scanpath Analysis study. The common scanpaths retrieved from this study consist of a list of visual elements which indicates their general viewing order. We detected the roles of visual elements in each common scanpath and tried to generate a common role path to apply on all web pages. Our work showed that, although the preliminary results of eMine Scanpath Analysis study are very promising, generalization of role path requires more comprehensive data.

Finally, we designed a transcoding system in a proxy-server architecture. To demonstrate this system, we then implemented it on ACTF platform, where we developed of visual element identification and role detection components. The proposed architecture consisted of two modes, which are block based transcoding and role based

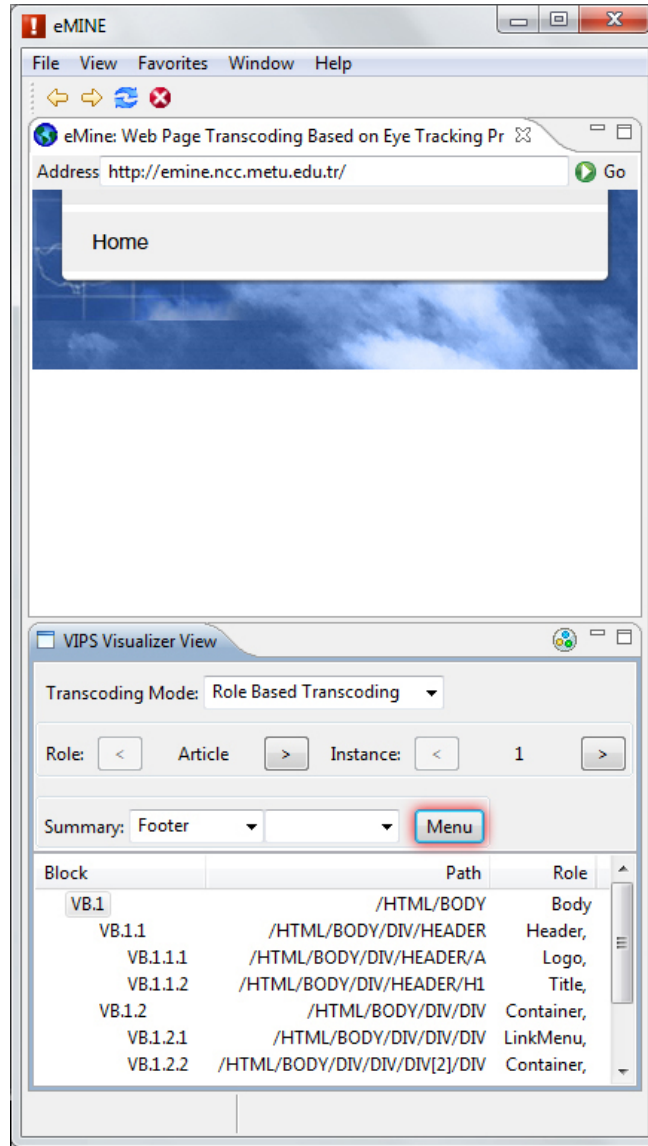


Figure 5.9: Direct access to menu

transcoding. Block based transcoding mode uses common scanpath of a single web page to transcode it. In role based transcoding, we used an example role path to navigate user through the visual elements in the web page. The simulation of this system showed that it is possible to transcode web pages by using eye tracking data as well as underlying roles of visual elements in a web page.

Our preliminary results with web page transcoding was promising; however, to validate the transcoding simulation, we conducted both user and technical evaluation. The next chapter explains the evaluation procedure and its results.

CHAPTER 6

EVALUATION

In the preceding chapters, we described our automatic role detection approach and web page transcoding approach based on eye tracking. In order to validate these approaches, we have conducted four evaluations. These evaluations include both technical and user evaluations separately for automatic role detection and web page transcoding. In this chapter, we first describe the evaluation methods and materials; then we present and discuss the results of these experiments.

This chapter is organized as following: First of all, Section 1.1 describes the user evaluation of our automatic role detection approach. Section 1.2 explains technical evaluation of this approach. Then, Section 1.3 moves on to describe user evaluation of our transcoding implementation based on eye tracking. Finally, Section 1.4 presents the technical evaluation of this implementation.

6.1 User Evaluation of Automatic Role Detection

To evaluate our work on web page segmentation and heuristic role detection, we have conducted a user experiment with an online survey. Online survey was selected as the experiment method to reach more people from different countries, levels of expertise, age groups, education and professional backgrounds. In this section, we present the evaluation procedure, our main research questions in this evaluation and the results of the user experiment.

Table 6.1: Web pages used in the user study and their complexity scores

Complexity	Pages
Low	http://home.mywebsearch.com (0.38), http://www.bing.com (0.33), http://www.apple.com (0.64), http://www.google.com.tr (1.1), http://adf.ly (1.43), http://imgur.com (1.58) http://wordpress.org (1.70), http://babylon.com (2.20), http://www.dailymotion.com/tr (2.40)
Medium	http://www.microsoft.com/en-us/default.aspx (3.08), http://www.avg.com/tr-tr/homepage (3.11), http://www.ebay.com (3.12), http://www.youtube.com (3.96), http://www.adobe.com (4.14), http://www.zedo.com (5.01), http://www.flickr.com (5.53), http://www.yahoo.com (6.43), http://www.alibaba.com (6.69), http://en.wikipedia.org/wiki/Main_Page (6.96)
High	http://www.aol.com (7.73), http://www.mediafire.com (7.74), http://www.craigslist.org/about/sites (8.57), http://ask.com (9.42), http://www.bbc.co.uk (9.93), http://www.uol.com.br (10), http://www.godaddy.com (10), http://www.about.com/#!/editors-picks (10), http://stackoverflow.com (10), http://www.huffingtonpost.com (10)

6.1.1 Materials

To select the web pages used in this experiment, we first listed the most popular 100 web pages from Alexa. By using the Visual Complexity Rankings and Accessibility Metrics (VICRAM) framework [90], we calculated a Visual Complexity Score (VCS) for each page. Then, we grouped these web pages into three: low complexity – pages with VCS lower than 3; medium complexity – pages with a VCS between 3 and 7; and high complexity – pages with a VCS above 7. Finally, we randomly selected ten web pages from each complexity group. However, when we were preparing the material for this experiment, one web page with low complexity was removed from the list, since it had dynamic content and we could not take screenshots properly. Therefore, a total of 29 pages were used in the experiment. Table 6.1 lists these selected pages with their complexity scores. With this approach, we applied a systematic method for choosing pages with different levels of complexity.

For each participant, a total of nine web pages including three pages from each complexity group were selected randomly. However, the participants were not informed about the complexity of web pages during the experiment.

Despite the majority of web pages whose top 5 levels of segmentation were used, some pages only had 3 or 4 levels due to their simple structures. For example, `home.mywebsearch.com/` had only 3 levels; and `www.bing.com/`, `www.google.com.tr/` and `babylon.com/` had only 4 levels. However, these cases are also very significant, since there are many pages which have simple structures. In our user evaluation, we calculated rating and ranking results of these web pages separately.

6.1.2 Administration

The experiment was conducted as an online survey. The survey application was developed using PHP based CodeIgniter framework. The data collected was stored in a MySQL database. The online study was released on the 7th of February 2013. The call for participation was distributed in several mailing lists which include CHI announcements, SIGWEB announcements, social media sites such as LinkedIn, Chamber of Computer Engineers in Turkey, and also sent to personal contacts and groups working on web design. Until 25th of July 2013, when the survey was closed for participation, 253 individuals participated in our study and 34 of them completed at least three pages in overall.

6.1.3 Procedure

The online survey included four main parts:

Information Sheet: First page in the experiment included an information sheet about the study. In this page we presented a brief introduction about the study, the tasks to be completed by participants and anonymity of the study.

Demographics: The second part included a survey to retrieve demographic information about the participants. We collected data about their gender, age, web usage frequency, current status involved in web design and development, level of expertise on web design and development and highest level of education completed.

Visual Elements Identification: In this part, participants were provided graphical representation of segmented visual elements in a randomly selected web page in different levels. Based on these representations, the participants were asked to rate given segmentation results. Afterwards, we asked the participants to rank these levels from 1 to 5. Here, 1 indicates the best level, while 5 indicates the worst level.

Role Assignment: Based on the best segmentation level they have chosen in the previous step, the participants were asked to assign roles to each visual element in the corresponding level. They were provided a list of roles; however, if the

participants could not find the proper role in the list, they were given the chance to specify the exact role of the visual element.

In overall, during the experiment, participants repeated visual element identification and role assignment steps for nine randomly selected web pages. The random function to select web pages was designed so that it returned pages from different complexity groups in each consecutive selection. The participants were free to leave the experiment anytime they wanted; however, we expected them to evaluate at least one page from each complexity group. Therefore, we omitted the participants who evaluated less than three pages to ensure the consistency of the data.

6.1.4 The Scientific Questions

In our user evaluation, we aimed to gather data to evaluate our proposed segmentation and heuristic role detection algorithms, based on a set of scientific questions. The main questions addressed in this section are as follows:

Research Question I How successfully does our visual element identifier divide web pages into its segments in a specific segmentation level?

Using the rating results in the first part, we aim to measure the success of our visual element identifier. In addition, this research question focus on the success in web pages with low, medium and high complexity; and for different demographic profiles. Moreover, this research question investigates the effect of visual complexity of web pages and demographic profiles of the participants on rating results, as well.

Research Question II Which level is preferred by the participants among the first 5 levels of segmentation?

Based on the ranking results, this question focus on which level is selected as the best level by the participants. The main intent of this research question is to specify the most preferred level of segmentation for later use of visual element identification in applications such as web page transcoding. Similar to Research Question I, this research question investigates best level selection for different visual complexity groups and different participant profiles. Moreover, this research question investigates the effect of visual complexity of web pages

Table 6.2: Demographic summary of participants

Criteria	Value	Count	Percent (%)
Gender	Female	14	41.18
	Male	20	58.82
Age	Under 18	0	0.00
	18-24	8	23.53
	25-34	13	38.24
	35-54	11	32.35
	55+	2	5.88
Web usage	Daily	34	100.00
	Weekly	0	0.00
	Monthly	0	0.00
	Less than once a month	0	0.00
	Never	0	0.00
Current status in web design and development	Worked	25	73.53
	Studied	4	11.76
	Hobby	4	11.76
	Other	1	2.94
Level of expertise	Professional	13	38.24
	Intermediate	16	47.06
	Novice/Beginner	5	14.71
Education	Grade/Primary School	0	0.00
	High/Secondary School	6	17.65
	Associate's Degree	2	5.88
	Bachelor's Degree	6	17.65
	Master's Degree	11	32.35
	Doctorate	9	26.47
	Other	0	0.00

and demographic profiles of the participants on ranking results.

Research Question III Is there a relation between the rating results and the ranking results?

To determine whether participants rate the levels based on their ranking preferences, we investigate a correlation between rating and ranking results of the participants.

Research Question IV What is the success of our heuristic role detection algorithm?

Here, we aim to calculate the success rate of our proposed approach by comparing the system generated roles of visual elements to the responses of participants who are provided a list of roles defined in our knowledge base.

6.1.5 Participants

The survey results include many participants with different profiles. Demographic distribution of the participants is given in Table 6.2. After eliminating the participants who evaluated less than three pages, we had 34 participants in overall.

6.1.6 Results

In overall, 327 pages have been rated including 102 pages with low complexity, 113 pages with medium complexity and 111 pages with high complexity. 202 of pages were considered as valid, since their assigners satisfied the minimum requirement of evaluating at least one page from each visual complexity group, including 68 low, 66 medium and 68 high complexity pages respectively. Moreover, 1,946 role assignments have been made and 1,458 were considered in our evaluation as valid assignments, due to the number of pages evaluated by their assigners. 232 roles assigned to pages with low complexity, 580 roles were assigned to pages with medium complexity and 646 roles were assigned to pages with high complexity.

During this evaluation process, three types of data were collected. First of all, the data retrieved in rating part consists of 5-point scale Likert items for top five levels of segmentation for each web page. Similarly, the data collected in ranking part includes the ranking results of each participant for 5 levels. The ranking values ranges from 1 to 5, where 1 indicates the best level and 5 indicates the worst level. Moreover, the data used in role detection evaluation includes the roles assignments of participants for each visual element and corresponding roles that our system generated for them. Finally, each participant had demographic data in six dimensions, which was used to group participants with respect to different demographic profiles. We performed statistical calculations by using IBM SPSS Statistics 21.

6.1.6.1 Techniques of Evaluation

To evaluate the data, a set of statistical methods were used. First of all, to evaluate the rating results, we calculated mean and standard deviation of each sample. In ranking part, we have calculated the frequencies and percentages of ranking results for all responses. To determine the effect of visual complexity of web pages over rating and ranking results, we applied Paired T Test, since there was dependence of observation between two samples. Finally, we used Pearson's Chi Square Test to measure the effect of demographic profiles, such as age or gender of the participants.

Table 6.3: Overall rating results

	3 Levels			4 Levels			5 Levels		
	N	Mean	Std. Dev.	N	Mean	Std. Dev.	N	Mean	Std. Dev.
Level 1	8	2.125	0.8345	23	3.217	1.5063	171	2.322	1.2398
Level 2	8	3.375	1.0607	23	3.217	1.0426	171	2.795	1.1320
Level 3	8	3.875	1.2464	23	3.609	1.3052	171	3.298	0.9695
Level 4	-	-	-	23	3.609	1.0762	171	3.509	1.0311
Level 5	-	-	-	-	-	-	171	3.637	1.1770

Table 6.4: Rating results for complexity groups with 5 levels

	Low Complexity			Medium Complexity			High Complexity		
	N	Mean	Std. Dev.	N	Mean	Std. Dev.	N	Mean	Std. Dev.
Level 1	37	2.405	1.2793	66	2.349	1.2215	68	2.250	1.2504
Level 2	37	2.945	1.1772	66	2.833	1.1580	68	2.677	1.0851
Level 3	37	3.513	1.0440	66	3.303	0.9762	68	3.177	0.9133
Level 4	37	3.405	0.9267	66	3.606	1.1077	68	3.471	1.0144
Level 5	37	3.621	1.0633	66	3.576	1.2286	68	3.706	1.1977

6.1.6.2 Rating Results for Visual Element Identification

The results of the perceived success of our visual element identification implementation, based on participant responses are given in Table 6.3. These results indicate that, segmentation in lower levels is rated as below average, while segmentation in middle or higher levels is rated as above average. Moreover, the success rate, in general, increases as parallel to the granularity of segmentation.

Table 6.4 presents the perceived success of our visual element identification implementation with respect to the complexity of the pages. According to these results, success rates increase while the level number increases, except for Level 4 of low complexity group and Level 5 of medium complexity group. Segmentation in lower levels is rated as below average while segmentation in middle and higher levels is rated as above average. To further investigate the significance of visual complexity, we applied Paired T Test on rating data. The results of these tests are presented in Table 6.5. According to these results, the complexity of a page is not a significant criteria for rating the segmentation levels. Another finding was that, for any complexity group, there is a clear trend of increasing in rating results with respect to the increasing in level number.

The results of the perceived success of our segmentation algorithm, grouped by different participant profiles, are presented in Appendix F.1. The results of this analysis

Table 6.5: Paired T Test results on rating data

	Paired Differences							
				95% Confidence Interval of Difference				
Pairs	Mean	Std. Dev.	Std. Err. Mean	Lower	Upper	t	df	Sig. (2-tailed)
Low - Med. (L. 1)	0.412	1.083	0.186	0.034	0.790	2.218	33	0.034
Low - Med. (L. 2)	0.289	0.949	0.163	-0.042	0.620	1.776	33	0.085
Low - Med. (L. 3)	0.162	1.022	0.175	-0.195	0.518	0.923	33	0.363
Low - Med. (L. 4)	-0.167	1.135	0.204	-0.583	0.250	-0.817	30	0.420
Low - Med. (L. 5)	0.139	1.469	0.300	-0.482	0.759	0.463	23	0.648
Low - High (L. 1)	0.314	0.921	0.158	-0.008	0.635	1.986	33	0.055
Low - High (L. 2)	0.338	0.872	0.150	0.034	0.642	2.262	33	0.030
Low - High (L. 3)	0.255	1.075	0.184	-0.120	0.630	1.383	33	0.176
Low - High (L. 4)	0.022	1.045	0.188	-0.362	0.405	0.115	30	0.910
Low - High (L. 5)	0.160	1.217	0.248	-0.354	0.674	0.643	23	0.527
Med. - High (L. 1)	-0.098	0.789	0.135	-0.373	0.177	-0.724	33	0.474
Med. - High (L. 2)	0.049	0.795	0.136	-0.228	0.326	0.359	33	0.722
Med. - High (L. 3)	0.093	0.829	0.142	-0.196	0.382	0.655	33	0.517
Med. - High (L. 4)	0.172	1.043	0.179	-0.192	0.536	0.959	33	0.345
Med. - High (L. 5)	-0.113	1.129	0.194	-0.507	0.281	-0.582	33	0.564

indicate that older participants rated our segmentation algorithm as more successful in middle levels, while younger participants rated our segmentation algorithm as more successful in higher levels. Moreover, female participants gave higher rating results than male participants for higher levels, while male participants gave higher rating results than female participants for lower levels. Among three groups in level of expertise, the lowest rating results belong to professional participants. A possible explanation for this might be that, professional participants have been involved in web design and development, and have knowledge and experience more than other participant groups. Therefore, they may have responded in more selective and critical approach. Finally, the survey conducted with the participants who have worked in web design and development, resulted in the lowest values of segmentation rating. A possible explanation for this is similar to the level of expertise criteria. It is somewhat surprising that, the rating results of the participants who have studied in this field are in the same range (between average and above average) for each level, unlike other results in which at least one level appears in a lower or higher range. It is difficult to explain this result, but it might be due to the fact that, this participant group examined segmentation process as a whole, rather than dividing the task to the levels separately. To investigate the effect of different participant demographic attributes, we conducted Pearson's Chi Square Tests for each criteria. The results of these tests are presented in Appendix F.2. As can be seen from this table, demographic profiles are significant

Table 6.6: Ranking results for pages with 3 and 4 levels

Level	Pages with 3 levels			Pages with 4 levels			
	Best Level		Worst Level	Best Level			Worst Level
1	1 (12%)	0 (0%)	7 (87,5%)	6 (26.1%)	2 (8.7%)	2 (8.7%)	13 (56.5%)
2	3 (37,5%)	4 (50,0%)	1 (12,5%)	0 (0%)	6 (26.1%)	16 (69.6%)	1 (4.3%)
3	4 (50,0%)	4 (50,0%)	0 (0%)	5 (21.7%)	12 (52.2%)	4 (17.4%)	2 (8.7%)
4	-	-	-	12 (52.2%)	3 (13.0%)	1 (4.3%)	7 (30.4%)

Table 6.7: Ranking results for pages with 5 and more levels

	Best Level				Worst Level
Level 1	12 (7,0%)	14 (8,2%)	13 (7,6%)	18 (10,5%)	114 (66,7%)
Level 2	23 (13,5%)	15 (8,8%)	16 (9,4%)	108 (63,2%)	9 (5,3%)
Level 3	22 (12,9%)	23 (13,5%)	112 (65,5%)	10 (5,8%)	4 (2,3%)
Level 4	31 (18,1%)	92 (53,8%)	19 (11,1%)	22 (12,9%)	7 (4,1%)
Level 5	83 (48,5%)	27 (15,8%)	11 (6,4%)	13 (7,6%)	37 (21,6%)

criteria for rating results of the participants in general.

6.1.6.3 Ranking Results for Visual Element Identification

The results of participant preference on level selection are given in Table 6.6 and Table 6.7. It is apparent from these tables that, participants selected highest available level as the best level, and lowest level as worst level. Moreover, the results of participant preference on level selection according to the page complexities, are given in Table 6.8. Similarly, for each complexity group, participants selected highest level as their best level, and lowest level as their worst level. Contrary to our expectations, this analysis did not find a significant difference between best level preferences for different complexity groups. To investigate the effect of visual complexity of a web page in best level preference, we conducted Paired T Tests to compare ranking results of participants for each complexity group. The results of these tests are presented in Table 6.9. According to these results, we can conclude that, there is no statistically significant difference between the results of two visual complexity groups and the differences between the means are likely due to chance and not likely due to the IV manipulation. This finding was unexpected and suggests that complexity is not a major factor in best level preference of the participants.

The results of the participant preference on level selection, grouped by different participant profiles, are given in Appendix F.3. Unlike other participant groups, partic-

Table 6.8: Ranking results for complexity groups

Compl.	Level	Best Level				Worst Level
Low	1	2 (5.4%)	4 (10.8%)	2 (5.4%)	6 (16.2%)	23 (62.2%)
	2	6 (16.2%)	4 (10.8%)	5 (13.5%)	21 (56.8%)	1 (2.7%)
	3	7 (18.9%)	7 (18.9%)	19 (51.4%)	3 (8.1%)	1 (2.7%)
	4	5 (13.5%)	20 (54.1%)	6 (16.2%)	4 (10.8%)	2 (5.4%)
	5	17 (45.9%)	2 (5.4%)	5 (13.5%)	3 (8.1%)	10 (27.0%)
Medium	1	4 (6.1%)	5 (7.6%)	8 (12.1%)	7 (10.6%)	42 (63.6%)
	2	10 (15.2%)	5 (7.6%)	6 (9.1%)	42 (63.6%)	3 (4.5%)
	3	8 (12.1%)	8 (12.1%)	45 (68.2%)	3 (4.5%)	2 (3.0%)
	4	18 (27.3%)	31 (47.0%)	4 (6.1%)	10 (15.2%)	3 (4.5%)
	5	26 (39.4%)	17 (25.8%)	3 (4.5%)	4 (6.1%)	16 (24.2%)
High	1	6 (8.8%)	5 (7.4%)	3 (4.4%)	5 (7.4%)	49 (72.1%)
	2	7 (10.3%)	6 (8.8%)	5 (7.4%)	45 (66.2%)	5 (7.4%)
	3	7 (10.3%)	8 (11.8%)	48 (70.6%)	4 (5.9%)	1 (1.5%)
	4	8 (11.8%)	41 (60.3%)	9 (13.2%)	8 (11.8%)	2 (2.9%)
	5	40 (58.8%)	8 (11.8%)	3 (4.4%)	6 (8.8%)	11 (16.2%)

ipants who completed bachelor's degree or doctorate have selected Level 1 as their best level and Level 4 as their worst level. Moreover, participants aged over 55 have selected Level 3 as their best level and Level 5 as their worst level. Finally, the participants who have studied web design and development have selected Level 2 as the best level and Level 5 as the worst level. It is difficult to explain this result, but it might be related to a simplistic approach of the participants who studied web design. We conducted Pearson's Chi Square Tests for each criteria to investigate the effect of different participant profiles on ranking results. The results of these tests are presented in Appendix F.4. According to these results, there was a significant positive correlation between best level preference of the participants and age, education and current status in web design and development criteria.

6.1.6.4 Correlation Between Rating and Ranking Results

When we analyzed rating results, we observed that the rating results increases from the 1st level through the 5th level. Moreover, according to the ranking results, participants ranked the 5th level as their best level preference. Based on this similarity, we investigated a correlation between rating and ranking results by performing Pearson's Chi Square Tests on the results. The results of these tests are given in Table 6.10 and Table 6.11. These results suggest that there is a relationship between rating and ranking results at 5% significance level.

Table 6.9: Pairwise T Test results on ranking data

	Paired Differences							
				95% Confidence Interval of Difference				
	Mean	Std. Dev.	Std. Err. Mean	Lower	Upper	t	df	Sig. (2-tailed)
High - Med. (L.1)	0.081	1.359	0.173	-0.264	0.426	0.467	61	0.642
High - Med. (L.2)	0.194	1.114	0.141	-0.089	0.476	1.368	61	0.176
High - Med. (L.3)	0.000	0.905	0.115	-0.230	0.230	0.000	61	1.000
High - Med. (L.4)	0.129	1.048	0.133	-0.137	0.395	0.970	61	0.336
High - Med. (L.5)	-0.403	1.273	0.162	-0.727	-0.080	-2.493	61	0.015
High - Low. (L.1)	0.000	1.495	0.253	-0.514	0.514	0.000	34	1.000
High - Low. (L.2)	0.411	1.523	0.204	0.003	0.819	2.018	55	0.048
High - Low. (L.3)	0.097	0.953	0.121	-0.145	0.339	0.799	61	0.427
High - Low. (L.4)	0.097	1.445	0.184	-0.270	0.464	0.527	61	0.600
High - Low. (L.5)	-0.210	1.747	0.222	-0.653	0.234	-0.945	61	0.348
Med. - Low (L.1)	-0.143	1.630	0.275	-0.703	0.417	-0.519	34	0.607
Med. - Low (L.2)	0.214	1.436	0.192	-0.170	0.599	1.117	55	0.269
Med. - Low (L.3)	0.095	1.027	0.129	-0.163	0.354	0.736	62	0.465
Med. - Low (L.4)	-0.032	1.459	0.184	-0.399	0.336	-0.173	62	0.863
Med. - Low (L.5)	0.190	1.674	0.211	-0.231	0.612	0.903	62	0.370

6.1.6.5 Evaluation Results of Automatic Role Detection

To evaluate the success of our automatic role detection approach, first we did strict string comparison between system generated roles and participant assignments. However, the overall results were very unsatisfactory due to a set of reasons, such as participants' diverse perception of roles, task independent and image based experiment, and indistinctive names of some roles. Therefore, we applied a receptive evaluation on the data, which consists of manual comparison. Table 6.12 presents the results obtained from the preliminary analysis of the data obtained from the participants who have worked in or studied web design and development as professional. The success rate of each complexity group and overall result was calculated proportional to the number of visual elements which were evaluated in each page. System - Expert evaluation consists of the comparison of the system results and expert responses with respect to the concept described in the ontology. As can be seen from this table, in overall the system has an accuracy of 86.83% accuracy. We then conducted a strict string comparison between the roles assigned by the system and also the participants. This gave us an average of 28.86% (low complexity pages - 26.32%, medium complexity - 28.99% and high complexity - 29.92%) accuracy. This is mainly because participants use slightly different versions of the role text used to label visual elements. Therefore, we have manually analyzed the role assignments given by our participants and

Table 6.10: Rating * ranking crosstabulation

		Best Level	2nd Level	3rd Level	4th Level	Worst Level	Total
Excellent	Count	70	23	14	7	5	119
	Expected Count	23,8	23,8	23,8	23,8	23,8	119,0
	% within Ranking	40,9%	13,5%	8,2%	4,1%	2,9%	13,9%
Above Average	Count	67	78	48	24	11	228
	Expected Count	45,6	45,6	45,6	45,6	45,6	228,0
	% within Ranking	39,2%	45,6%	28,1%	14,0%	6,4%	26,7%
Average	Count	29	49	66	57	35	236
	Expected Count	47,2	47,2	47,2	47,2	47,2	236,0
	% within Ranking	17,0%	28,7%	38,6%	33,3%	20,5%	27,6%
Below Average	Count	3	19	38	61	53	174
	Expected Count	34,8	34,8	34,8	34,8	34,8	174,0
	% within Ranking	1,8%	11,1%	22,2%	35,7%	31,0%	20,4%
Extremely Poor	Count	2	2	5	22	67	98
	Expected Count	19,6	19,6	19,6	19,6	19,6	98,0
	% within Ranking	1,2%	1,2%	2,9%	12,9%	39,2%	11,5%
Total	Count	171	171	171	171	171	855
	Expected Count	171,0	171,0	171,0	171,0	171,0	855,0
	% within Ranking	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%

Table 6.11: Chi-Square Test results

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	433.070	16	0
Likelihood Ratio	420.163	16	0
N of Valid Cases	855		

compared them with our system assignments. These results are shown as "receptive evaluation" in Table 6.12. In overall, the accuracy rate is 80.82%.

6.1.7 Discussion

In Section 6.1, we presented our user evaluation on visual element identification and automatic role detection. The rating results of visual element identification indicate that, although the rating results for lower levels stayed below average, middle and higher levels were rated as above average. This finding was unexpected, because the same segmentation rules apply on each visual element in web pages. Therefore, we expected the rating results of a web page to be similar in all levels. Moreover, partici-

Table 6.12: Accuracy results of role detection algorithm

Complexity Group	System-Expert Evaluation	Receptive Evaluation
Low Comp.	79.82	73.68
Medium Comp.	88.28	79.77
High Comp.	88.47	85.53
Overall	86.83	80.82

pants preferred higher levels as their best level of segmentation than lower levels. This result may be related to that, higher levels have more detailed segmentation and simpler visual elements, while lower levels are consisted of larger visual elements. These findings also have important implications for use of segmentation process in further studies, such as developing a transcoding application based on web page segmentation. Finally, our further investigation showed that, there is a positive correlation between rating and ranking results. This means that, the responses of the participants are consistent in terms of both rating and ranking results.

Our receptive evaluation on automatic role detection showed that, our proposed system has more than 80% accuracy rate. However when we did strict string comparison, this accuracy was around 30%. A possible explanation for these results is that, people perceive the roles of visual elements in different ways. This finding confirms the majority rule. When the majority rule is applied to role assignments, we see that only %32.58 of them have the majority of participants' agreements. There are several possible explanations for this result. First of all, the role assignment experience was not associated with specific tasks; therefore, participants assigned roles in different perspectives. A task based experiment could have been resulted in agreement of the majority of the participants. Moreover, in this experiment, although participants were encouraged to check the underlying source code of web pages and associated CSS files; the main role assignment tasks were performed on given screenshots of web pages. Therefore, participants may not have analyzed DOM structure or CSS styles of web pages and they formed their overall assessment only on the visual representation of the visual elements. Finally, some roles in the selection list could have been perceived very similarly and a descriptive explanation of such roles were not provided. Thus, participants may not have comprehended the difference.

One unanticipated finding was that, some visual elements may have multiple roles in web pages. However, in this experiment, participants were permitted to assign only one role. To overcome this, some participants typed multiple roles in the text area, which was for the roles that do not appear in the selection list. Moreover, some visual elements are combinations of smaller visual elements that stand for different purposes. In our terminology, we refer them as *Container*. However, when we look at the roles assigned by the participants, we have noticed that some participants assigned

Table 6.13: Performance results for visual element identification

	Complexity	High	Medium	Low
Dom Structure Cons.	Memory Usage	20,659.53 KB	16,293.37 KB	9,370.56 KB
	Time	5,069 ms	2,120 ms	730 ms
Visual Block Extr.	Memory Usage	21,610.17 KB	14,351.68 KB	9,535.14 KB
	Time	30 ms	19 ms	4 ms
Content Structure Cons.	Memory Usage	21,519.97 KB	13,711.88 KB	9,699.04 KB
	Time	93 ms	53 ms	21 ms
Total Node Count		3,811	1,888	624
Max. Node Depth		22	21	16
Visited Node Count		2,609	1,328	395
Invalid Node Count		98	84	44
Invisible Node Count		69	35	26
Visual Element Count		569	237	65
Visual Element Depth		13	11	7

the role of an interior visual element to the *Container* element and omitted the roles of other visual elements inside it. This issue was addressed by manually comparing the roles, which participants and the system assigned.

6.2 Technical Evaluation of Automatic Role Detection

To investigate the technical feasibility of proposed approaches and their implementation, we also conducted a technical evaluation. In this process, we mainly focused on the performance characterized in terms of total memory usage and response time, and presented the results with total number of visual elements in the web pages. The technical evaluation has been performed on a machine which has following features: Intel®Core™2 Duo CPU T9600 @ 2.80 GHz processor, 2,071.34 MB memory, NVIDIA GeForce GT 220M video card and Windows 7 32 Bit operating system. Even though, we understand the results presented in this section are specific to our configuration of test machine, they still provide significant information about the overall performance of the system.

We evaluated visual element identification and automatic role detection components separately. Table 6.13 presents the performance results of the visual element identification component. Moreover, the performance results of automatic role detection component are shown in Table 6.14.

Table 6.14: Performance results for automatic role detection

Complexity Group	Total Memory	Total Time	Average Memory per Visual Element	Average Time per Visual Element	Visual Element Count
Low	8,369 KB	6,576 ms	244.29 KB	102.29 ms	65
Medium	7,013 KB	23,799 ms	36.44 KB	102.12 ms	237
High	9,165 KB	54,837 ms	34.28 KB	101.95 ms	569
Overall	8,176 KB	29,157 ms	100.20 KB	102.11 ms	298

6.2.1 Discussion

According to the results of technical evaluation of visual element identification, DOM structure construction step uses higher amount of memory than other steps and executes in longer time. The main reason for this is that, all nodes in a web page are traversed and converted to a proper format for use in later steps. Therefore, the total memory usage and time are related with the casting operation. Reducing this operation may result in better performance results. Since node count changes with respect to the visual complexity of web pages, the performance of the system is higher in web pages with low complexity. Therefore, the node count of a web page is a major criterion in visual element identification. While node count increases, memory usage and total time elapsed also increase.

Average time elapsed for automatic role detection of a single visual element has close values in each visual complexity group. Moreover, total time elapsed for automatic role detection of the whole web page is proportional to the number of visual elements in web pages. Total memory consumed for overall web page has also close values for each visual complexity group, showing that, larger amount of the memory consumed for shared static resources, such as working memory. Therefore, average memory consumed decreases while the number of visual elements increases.

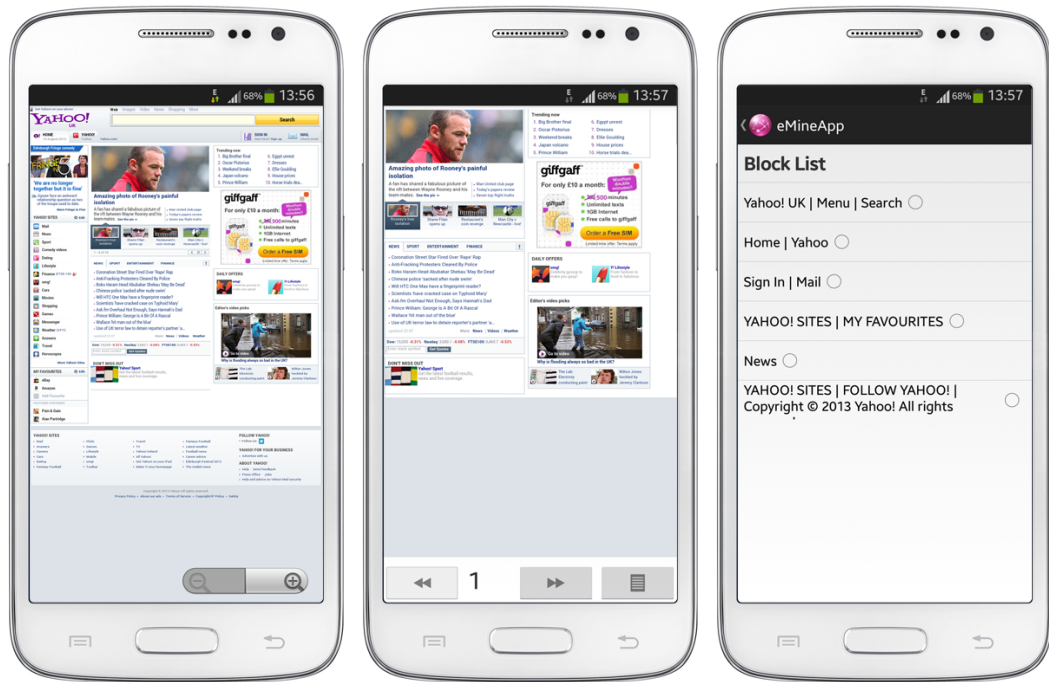
It is important to note that, in our study we examined overall web page in both visual element identification and automatic role detection. In a typical application in which these components are deployed, visual elements in web pages would be examined until a certain level of granularity is reached. Therefore, this would reduce the response time and memory usage to a reasonable level.

6.3 User Evaluation of Web Page Transcoding Based on Eye Tracking

In the previous sections, we evaluated our visual element identification implementation and automatic role detection approach. Based on these components, we have developed a transcoding application based on eye tracking. In this section, we aim to evaluate this application in terms of task completion, system usability, information quality and interface quality. To determine the difference in user satisfaction in original and transcoded pages, we conducted a user evaluation in which, two user groups perform specific tasks on and interact with them. During the user experiments, we applied both subjective and objective evaluation methods. In subjective evaluation, we conducted a survey in which, participants evaluate the system based on their performance. In objective evaluation, participants have been observed during their performance to better interpret the results.

6.3.1 Materials

For continuity of the study, we used six web pages which were used in our previous studies. These web pages are grouped into three based on their visual complexities including low, medium and high complexity pages. In our previous experiments, it was shown that 5th segmentation level is the most preferred by the participants; therefore, we used visual elements in the 5th level in transcoding of the pages. Two different software to simulate both transcoded and original versions of the pages were developed for this study. The software which represented original version of the pages consisted of an interface enabling the user to zoom in and zoom out with pinch gesture (Figure 6.1a). The other software, which represented a transcoded version of the page included three buttons (Figure 6.1b); two of them were used to move across the elements in scanpath and the third button was used to retrieve a list of blocks in the page in a summary form (Figure 6.1c). In each state in the user's travel across the page, a single visual element was represented. Users were also enabled to zoom in and zoom out with pinch gesture.



(a) Original page

(b) Transcoded Page

(c) Page Summary

Figure 6.1: Original and transcoded pages used in the study

6.3.2 Administration

The evaluation was conducted as personal interview survey. The survey was conducted in engineering departments of METU, METU Library and METU Teknokent. Graduate and undergraduate students and professional software developers were involved in the study with voluntary participation.

6.3.3 Procedure

The participants who agreed to participate in the study were given four forms:

Informed Consent Form This form included information about who the researcher is, the aim of the study in brief, assurance of the fact that, participation is on voluntary basis and that participants are free to withdraw the survey at any time, assurance of confidentiality of the data collected and a signed statement of agreement to participate voluntarily. At this phase, participants were informed

Table 6.15: Searching tasks used in the study

Apple	1. Can you locate a link which allows watching the TV ads relating to iPad mini?
	2. Can you locate a link labelled: iPad on the main menu?
Babylon	1. Can you locate a link you can download the free version of Babylon?
	2. Can you find and read the names of other products of Babylon?
Yahoo	1. Can you read the titles of the main headlines which have smaller images?
	2. Can you read the first item under News title?
AVG	1. Can you locate a link which you can download a free trial of Internet Security 2013?
	2. Can you locate a link which allows you to download AVG Anti-virus FREE 2013?
GoDaddy	1. Can you find a telephone number for technical support and read it?
	2. Can you locate a text box where you can search a new domain?
BBC	1. Can you read the first item of Sport News?
	2. Can you locate the table that shows market data under Business title?

that they would perform some tasks on a web page; however, whether the page is in original form or transcoded was not declared.

Demographic Information Questionnaire This form included a survey to retrieve demographic information about the participants. We collected gender, age, web usage frequency, web usage frequency in a mobile device, current status involved in web design and development, level of expertise on web design and development and highest level of education completed.

User Experience Survey Before the participants were given user experience survey form, they were randomly assigned to a web page and a representational form (original or transcoded). Participants were provided a Samsung Galaxy S3 I9300 phone with 4,8" 720 x 1 280 (HD) screen for the experiment and the usage of the software was introduced. Then, they were asked to perform some tasks using the software.

In this experiment, we used the tasks which Eraslan et al. applied in their eye tracking studies [42]. The tasks were divided into two: browsing and searching tasks. Browsing tasks simulated a web user who browses on a random web page. For browsing tasks, participants were asked to scan the web page if they find something interesting for them. Searching tasks, on the other hand, differed with respect to pages and simulated a web user who searches a specific content in the web pages. These tasks are represented in Table 6.15.

For objective evaluation, participants were observed during the experiment. Their comments and actions on the tasks were noted.

After participants completed tasks, they were asked to fill a questionnaire about

their experience. The questionnaire consisted of two main parts:

1. The After-Scenario Questionnaire (ASQ)
2. The Post-Study System Usability Questionnaire (PSSUQ) [76]

The questionnaire included 22 Likert scale items with 7-point scales. They were anchored with the terms "Strongly agree" for 1 and "Strongly disagree" for 7. Also, there was "Not Applicable" option outside the scale. ASQ consisted of Item 1 – Item 3, while PSSUQ consisted of Item 4 – Item 22. PSSUQ has three sub-parts: system usefulness (SYSUSE, Item 4 – Item 11), information quality (INFOQUAL, Item 12 – Item 18) and interface quality (INTERQUAL, Item 19 – Item 22). All items are listed in Appendix H.

Debriefing Form Finally, the participants were provided a debriefing form which explains the main purpose of the study and why participants were asked to complete some tasks on a particular web page.

6.3.4 Hypothesis

Supporting our proposal for a better way to represent web pages, the hypotheses of this study can be listed as following:

Hypothesis I: Participants who performed tasks on transcoded pages are more satisfied with their performance and system, in terms of task completion, system usability, information quality and interface quality.

Hypothesis II: Web page complexity affects the participant responses in task completion, system usability, information quality and interface quality questionnaires.

Hypothesis III: Demographic profiles, including gender, age, web usage in desktop and mobile environment, current status involved in web design and development, level of expertise and highest level of education completed, affect the participant responses in task completion, system usability, information quality and interface quality questionnaires.

Table 6.16: Demographic summary of participants

Criteria	Value	Count	Percent (%)
Gender	Female	25	50.0
	Male	25	50.0
Age	Under 18	0	0.0
	18-24	24	48.0
	25-34	25	50.0
	35-54	1	2.0
	55+	20	0.0
Web usage	Daily	50	100.0
	Weekly	0	0.0
	Monthly	0	0.0
	Less than once a month	0	0.0
	Never	0	0.0
Web usage in Mobile Environment	Daily	41	82.0
	Weekly	4	8.0
	Monthly	0	0.0
	Less than once a month	1	2.0
	Never	4	8.0
Current status in web design and development	Worked	11	22.0
	Studied	6	12.0
	Hobby	10	20.0
	Other	23	46.0
Level of expertise	Professional	9	18.0
	Intermediate	7	14.0
	Novice/Beginner	34	68.0
Education	Grade/Primary School	0	0.0
	High/Secondary School	13	26.0
	Associate's Degree	0	0.0
	Bachelor's Degree	30	60.0
	Master's Degree	7	14.0
	Doctorate	0	0.0
	Other	0	0.0

Hypothesis I is deduced from our major goal to propose a transcoding method to better display web pages on mobile devices as well as for disabled people. Moreover, we suspect that, visual complexity of a web page affects the page composition and layout of the web page, as well as, overall size of the page. Hence, Hypothesis II is inferred by our consideration of visual complexity of a web page as a major factor in users' web experience. Finally, our previous research showed that, people with different demographic profiles perceived the success of web page segmentation; therefore, Hypothesis III inferred by this finding.

6.3.5 Participants

The survey results includes many participants with different profiles. Demographic distribution of the participants is given in Table 6.16. These distributions consisted of

gender, age, web usage frequency, web usage frequency in a mobile device, current status involved in web design and development, level of expertise on web design and development, and highest level of education completed.

6.3.6 Results

The data used in the evaluation process consisted of dependent variables and an independent variable. Dependent variables were represented in four columns, each corresponding to the overall result of a participant for ASQ, SYSUSE, INFOQUAL and INTERQUAL tests. These results were calculated as mean values of ordinal variables as 7-point scale Likert items for a particular test. Independent variable consisted of two categorical independent groups as original pages and transcoded pages. Moreover, there was independence of observations between two groups, which means that, there were different participants in each group with no participant being in more than one group. Finally, each participant had demographic data in seven dimension, which helped to group participants with respect to different demographic attributes.

6.3.6.1 Techniques of Evaluation

In order to reduce 22-item questionnaire results to ASQ, SYSUSE, INFOQUAL and INTERQUAL tests, we calculated mean value of Item 1–Item 3, Item 4–Item 11, Item 12–Item 18 and Item 19–Item 22 in Appendix H in corresponding order. These calculations constructed our dependent variable sets for later statistical tests.

In order to investigate the perceived success of transcoded web pages and original web pages, we first calculated mean and standard deviation for ASQ, SYSUSE, INFOQUAL and INTERQUAL separately for our two independent groups. Then, to prove whether the difference between the samples of two groups is statistically significant, we conducted further tests. One of the techniques for determining whether there is a statistically significant difference between the means of two groups is independent t test. However, this test has a prerequisite: data must be normally distributed. Since our sample sizes are up to 25, Shapiro-Wilk Test is more appropriate. Then, we applied independent t test if data is normally distributed and a non-parametric test,

Mann-Whitney U Test if it does not. For specific participant groups, such as male participants, we filtered the data so that, we only analyzed a particular subset of overall data.

In order to determine whether there is a correlation with two result set, we conducted Pearson's Chi Square test for these groups.

6.3.6.2 Overall Test Results

Since the system did not give any error messages or provide online documentation, a minority of participants responded to Item 12 ("The system gave error messages that clearly tell me how to fix problems.", %24), Item 13 ("Whenever I made a mistake using the system, I could recover easily and quickly.", %42) and Item 14 ("The information (such as on-line help, on-screen messages and other documentation) provided with this system is clear.", %48). Many of the participants accepted the information in informed consent form and instructions we gave as support information for Item 3 ("Overall, I am satisfied with the support information (on-line help, messages, documentation) when completing this task.") and Item 14.

The results of the item based statistical analysis are presented in Table I.1 in Appendix I. Inspection of Q-Q Plots revealed that results for items 3, 5 ("It is simple to use this system."), 8 ("I was able to efficiently complete the tasks and scenarios using this system."), 12, 13, 14, 15 ("It was easy to find the information I needed.") and 19 ("The interface of this system was pleasant.") were normally distributed for both original and transcoded pages, while results for items 1 ("Overall, I am satisfied with the ease of completing this task."), 2 ("Overall, I am satisfied with the amount of time it took to complete this task."), 4 ("Overall, I am satisfied with how easy it is to use this system."), 6 ("I could effectively complete the tasks and scenarios using this system."), 7 ("I was able to complete the tasks and scenarios quickly using this system."), 9 ("I felt comfortable using this system."), 10 ("It was easy to learn to use this system."), 11 ("I believe I could become productive quickly using this system."), 16 ("The information provided for the system was easy to understand."), 17 ("The information was effective in helping me complete the tasks and scenarios."), 18 ("The organization of information on the system screens was clear."), 20 ("I liked using the

interface of this system."), 21 ("This system has all the functions and capabilities I expect it to have.") and 22 ("Overall, I am satisfied with this system.") were not.

Therefore, an independent t-test on items 3, 5, 8, 12, 13, 14, 15 and 19; and a Mann-Whitney U Test on items 1, 2, 4, 6, 7, 9, 10, 11, 16, 17, 18, 20, 21 and 22 were run on the data as well as 95% confidence intervals (CI) for the mean difference. It was found that, transcoded pages was not statistically significantly higher than the original pages in Item 3 ($t(30) = 0.142$, $p = 0.888$), Item 5 ($t(48) = 0.386$, $p = 0.701$), Item 8 ($t(48) = 1.443$, $p = 0.158$), Item 12 ($t(10) = 2.188$, $p = 0.054$), Item 13 ($t(19) = 0.359$, $p = 0.723$), Item 14 ($t(22) = 0.335$, $p = 0.741$), Item 15 ($t(48) = 1.662$, $p = 0.103$) and Item 19 ($t(48) = 0.592$, $p = 0.557$). Moreover, in all items except for Item 17, test results for transcoded pages had the highest results. However, in any item, transcoded pages were not statistically significantly higher than the original pages (Item 1 : $U = 245$, $p = 0.162$; Item 2 : $U = 277$, $p = 0.435$; Item 4 : $U = 249.5$, $p = 0.190$; Item 6 : $U = 309.5$, $p = 0.949$; Item 7 : $U = 267.5$, $p = 0.486$; Item 9 : $U = 308.5$, $p = 0.934$; Item 10: $U = 243$, $p = 0.128$; Item 11: $U = 233.5$, $p = 0.240$; Item 16: $U = 238.5$, $p = 0.261$; Item 17: $U = 283$, $p = 0.709$; Item 18: $U = 255$, $p = 0.336$; Item 20: $U = 266$, $p = 0.350$; Item 21: $U = 263$, $p = 0.594$; Item 22: $U = 278.5$, $p = 0.481$).

These results show that, there is no statistically significant difference between the responses to the items in the questionnaire for both original and transcoded web pages. Since the majority of the participants considered Item 12, 13 and 14 as inapplicable for the study, it was decided to dismiss these items from INFOQUAL test results. From now on, INFOQUAL results are calculated based on only Items 15, 16, 17 and 18.

6.3.6.3 Task Completion, System Usability, Information and Interface Quality Test Results

The aim of this study is to investigate the success of transcoded pages over original pages, based on participant responses to four usability tests: ASQ, SYSUSE, INFOQUAL and INTERQUAL. In this section, we compared the results for transcoded pages and original pages for all web pages and participant groups. Firstly, we calculated the mean and standard deviation of each test for both original and transcoded

Table 6.17: Test results for original and transcoded pages

	Original			Transcoded		
	N	Mean	Std. Dev.	N	Mean	Std. Dev.
ASQ	25	6.113	0.829	25	6.300	0.849
SYSUSE	25	5.846	1.121	25	6.195	0.735
INFOQUAL	25	5.890	1.075	25	6.300	0.997
INTERQUAL	25	5.650	1.337	25	5.760	1.448

pages. The results are represented in Table 6.17. However, in order to prove significance of these results, we conducted further tests on the data.

Inspection of Q-Q Plots revealed that ASQ, SYSUSE, INFOQUAL and INTERQUAL test results were not normally distributed for both original and transcoded pages. Therefore, a Mann-Whitney U Test was run on the data as well as 95% confidence intervals (CI) for the mean difference. It was found that, ASQ, SYSUSE, INFOQUAL and INTERQUAL test results for transcoded pages had the highest results; however, the results for transcoded pages were not statistically significantly higher than the results for original pages in ASQ ($U = 264$, $p = 0.334$), SYSUSE ($U = 267.5$, $p = 0.382$), INFOQUAL ($U = 217.5$, $p = 0.061$), INTERQUAL ($U = 290.5$, $p = 0.667$).

In conclusion, although the participants who performed tasks on transcoded pages seems to agree more strongly with the items in the questionnaire than those who performed on original pages, both participant groups gave similar responses to all usability tests.

6.3.6.4 Complexity Based Test Results and Comparison

Having compared the results of original and transcoded pages, we will now move on to compare these results with respect to the visual complexity of the web pages. In this process, we filtered the results for three visual complexity groups and calculated mean and standard deviation for both transcoded and original versions. The results are represented in Table 6.18.

Then, we investigated a statistical significance between the results of transcoded and original pages for different visual complexity groups. The results of corresponding tests are presented in Table 6.19. In overall, statistical investigation revealed that, both participants performed tasks on original and transcoded pages gave similar responses

Table 6.18: Test results for visual complexity groups

		Original			Transcoded		
		N	Mean	Std. Dev.	N	Mean	Std. Dev.
Low Complexity	ASQ	9	6.167	0.950	9	6.444	0.527
	SYSUSE	9	5.653	1.540	9	6.222	0.515
	INFOQUAL	9	5.639	1.541	9	6.472	0.690
	INTERQUAL	9	5.444	1.540	9	6.028	1.135
Medium Complexity	ASQ	8	6.208	0.689	8	6.458	0.582
	SYSUSE	8	6.031	0.853	8	6.484	0.403
	INFOQUAL	8	5.969	0.749	8	6.563	0.438
	INTERQUAL	8	5.563	1.499	8	5.969	0.940
High Complexity	ASQ	8	5.958	0.899	8	5.979	1.280
	SYSUSE	8	5.877	0.887	8	5.875	1.084
	INFOQUAL	8	6.094	0.743	8	5.844	1.523
	INTERQUAL	8	5.969	0.995	8	5.250	2.096

Table 6.19: Test results for visual complexity groups

	Low Complexity		Medium Complexity		High Complexity	
ASQ	Not Normal	U = 36, p = 0.682	Normal	t(14) = 0.784, p = 0.446	Normal	t(14) = 0.970, p = 0.038
SYSUSE	Not Normal	U = 38, p = 0.824	Normal	t(9.985) = 1.359, p = 0.204	Normal	t(14) = 0.996, p = 0.005
INFOQUAL	Not Normal	U = 27, p = 0.224	Normal	t(14) = 1.935, p = 0.073	Not Normal	U = 27.5, p = 0.633
INTERQUAL	Not Normal	U = 29, p = 0.303	Normal	t(11.767) = 0.650, p = 0.528	Not Normal	U = 29.5, p = 0.791

to items in all usability tests.

6.3.6.5 Correlation Between Test Results and Complexity

In the previous section, it was found that there was no statistically significant difference between test results of transcoded and original pages for all visual complexity groups. This raises questions about whether visual complexity of web pages is a significant criteria in the perceived success of transcoded and original pages. In order to investigate this, we conducted Pearson's Chi Square Test on the data. The results are represented in Table 6.20.

It is apparent from Table 6.20 that, there is no significant correlation at the $p = 0.05$ level. This concludes that, contrary to our expectations, web visual complexity of web pages did not effect the responses of participants in task completion, system usability, information quality and interface quality.

Table 6.20: Correlation with visual complexity

	Original	Transcoded
ASQ	X(6) = 6.354, p = 0.385	X(6) = 4.043, p = 0.671
SYSUSE	X(8) = 5.903, p = 0.658	X(6) = 6.096, p = 0.413
INFOQUAL	X(8) = 5.893, p = 0.659	X(8) = 8.064, p = 0.427
INTERQUAL	X(8) = 4.080, p = 0.850	X(10) = 7.473, p = 0.680

6.3.6.6 Demographics Based Test Results and Comparison

The participants of this study consist of different demographic groups and it is also worth comparing the usability test results based on these groups. In this section, for each demographic group, we compare means and investigate a statistically significant difference between original and transcoded pages. Appendix I.2 presents the mean and standard deviation of both original and transcoded web pages for different participant groups.

To investigate a statistical significance between the results of transcoded and original pages for different participant groups, we applied further statistical tests on the data. Appendix I.3 presents the results of these tests. According to these results, female and male participants performed tasks on original and transcoded pages gave similar responses to items in all usability tests. Similarly, participants who aged between 18–24 and 25–35 also gave similar responses to items in all usability tests. Since all of the participants in the study uses web daily, we did not conduct further analysis on the data for web usage criteria. For web usage in mobile devices, although it seems that test results for original and transcoded pages have similar values for each group, it is not possible to draw conclusion based on these data, since the majority of the participants (92%) are sampled in only one group. Moreover, for the participants who studied web design and development, there is a difference between the test results in ASQ and INFOQUAL, in which, those who performed tasks on transcoded pages gave higher responses than the others. Similarly, for the professional participants, there is also a difference between the test results in INFOQUAL, in which, those who performed tasks on transcoded pages gave higher responses than the others. Finally, there is a difference between the test results in INFOQUAL for the participants who completed bachelor's degree, in which, those who performed tasks on transcoded pages gave higher responses.

Table 6.21: Correlation with demographics

		Original	Transcoded
Gender	ASQ	X(3) = 3.287, p = 0.349	X(3) = 1.157, p = 0.763
	SYSUSE	X(4) = 1.777, p = 0.777	X(3) = 1.100, p = 0.777
	INFOQUAL	X(4) = 3.017, p = 0.555	X(4) = 1.867, p = 0.760
	INTERQUAL	X(4) = 1.777, p = 0.777	X(5) = 6.917, p = 0.227
Age	ASQ	X(6) = 6.964, p = 0.324	X(3) = 0.818, p = 0.845
	SYSUSE	X(8) = 5.317, p = 0.723	X(3) = 0.810, p = 0.847
	INFOQUAL	X(8) = 6.637, p = 0.576	X(4) = 2.292, p = 0.682
	INTERQUAL	X(8) = 3.810, p = 0.874	X(5) = 6.019, p = 0.304
Web Usage in Mobile Devices	ASQ	X(6) = 3.229, p = 0.780	X(9) = 3.540, p = 0.939
	SYSUSE	X(8) = 1.600, p = 0.991	X(9) = 9.805, p = 0.367
	INFOQUAL	X(8) = 4.236, p = 0.835	X(12) = 16.994, p = 0.150
	INTERQUAL	X(8) = 1.887, p = 0.984	X(15) = 28.095, p = 0.021
Current Status Involved in Web Design	ASQ	X(9) = 5.284, p = 0.809	X(9) = 6.585, p = 0.680
	SYSUSE	X(12) = 6.943, p = 0.861	X(9) = 9.144, p = 0.424
	INFOQUAL	X(12) = 8.746, p = 0.724	X(12) = 9.740, p = 0.639
	INTERQUAL	X(12) = 8.041, p = 0.782	X(15) = 7.176, p = 0.953
Level of Expertise	ASQ	X(6) = 5.717, p = 0.456	X(6) = 2.514, p = 0.867
	SYSUSE	X(8) = 2.461, p = 0.964	X(6) = 5.338, p = 0.501
	INFOQUAL	X(8) = 5.362, p = 0.718	X(8) = 4.596, p = 0.800
	INTERQUAL	X(8) = 16.330, p = 0.038	X(10) = 4.462, p = 0.924
Education	ASQ	X(6) = 7.443, p = 0.282	X(6) = 10.401, p = 0.109
	SYSUSE	X(8) = 9.712, p = 0.286	X(6) = 11.720, p = 0.069
	INFOQUAL	X(8) = 5.745, p = 0.676	X(8) = 12.336, p = 0.137
	INTERQUAL	X(8) = 10.735, p = 0.217	X(10) = 26.306, p = 0.003

6.3.6.7 Correlation Between Test Results and Demographics

This section investigates a relation between the results of usability tests and demographics. Comparisons between each test were made using Pearson's Chi Square Test. The results of the correlation analysis are summarized in Table 6.21.

It is apparent from this table that, there was not a significant positive correlation between usability test results and demographics at the $p = 0.05$ level; except for web usage in mobile devices and education in INTERQUAL for transcoded pages, as well as level of expertise in INTERQUAL for original pages.

6.3.6.8 Correlation Between Tests

The questionnaire used in the study consists of 22 items and these items can be divided into four different usability tests. In this section, we investigate a relation between the responses to different usability tests. Comparisons between each test were made using Pearson's Chi Square test. The results of the correlational analysis are

Table 6.22: Correlation between tests

	Original	Transcoded
ASQ – SYSUSE	X(12) = 43.785, p = 0.000	X(9) = 44.444, p = 0.000
ASQ – INFOQUAL	X(9) = 21.806, p = 0.010	X(12) = 31.213, p = 0.002
ASQ – INTERQUAL	X(12) = 14.236, p = 0.286	X(15) = 44.598, p = 0.000
SYSUSE – INFOQUAL	X(12) = 41.414, p = 0.000	X(12) = 40.596, p = 0.000
SYSUSE – INTERQUAL	X(16) = 31.566, p = 0.011	X(15) = 41.644, p = 0.000
INFOQUAL – INTERQUAL	X(12) = 23.063, p = 0.027	X(20) = 49.155, p = 0.000

summarized in Table 6.22.

It is apparent from this table that, there was a significant positive correlation between each test at the $p = 0.05$ level, except for ASQ and INTERQUAL tests applied to participants who performed tasks in original pages.

6.3.7 Discussion

The items used in this study were set out with the aim of assessing the success of our transcoding method in task completion, system usability, information quality and interface quality. In this section, we discuss each of these different aspects of the system.

Hypothesis I states that, participants who performed tasks on transcoded pages are satisfied more with their performance and system, in terms of task completion, system usability, information quality and interface quality. Although the results for transcoded pages have higher means than original pages, there is no statistically significant difference between the responses of two participant groups. Therefore, the data does not support this hypothesis. Moreover, Hypothesis II states that, visual complexity of web pages affects the participant responses in task completion, system usability, information quality and interface quality questionnaires. It was shown that, there is no statistically significant correlation between visual complexity and participant responses; thus, this hypothesis is not supported with the data. Finally, Hypothesis III states that, demographic profiles affect the participant responses in task completion, system usability, information quality and interface quality questionnaires. Similar with visual complexity, any of the demographic attributes is not statistically significantly correlated with participant responses. As a result, data does

not support this hypothesis. Taken together, these results show that, participants gave similar responses to the items in the questionnaire and the differences between the means are likely due to chance.

The items in ASQ were related with the satisfaction of the participants with the ease of and amount of the time it took to complete the tasks. The results of this study did not show any significant difference between original pages and transcoded pages in general. Only for the participants who studied web design and development, the results indicate that, the participants who performed tasks on a transcoded page were satisfied more than those who performed tasks on the original pages. Although the majority of the participants completed the tasks in satisfactory periods in both original and transcoded pages, some of them could not complete browsing tasks in original pages. Interestingly, one of the participants who never uses web on mobile environment, completed one of the tasks in relatively longer time in original page. However, she indicated that, for a user who never used web on mobile, it was the best that she could perform; therefore, she felt satisfied with her performance. Similar to this participant, the others who could not complete some tasks were satisfied with their performance; thus they responded to the questionnaire accordingly. It may be that the participants who performed tasks on original pages thought that, they were supposed to perform well since they were familiar with the representation. On the other hand, one of these participants commented that, the page was originally designed for large screens, therefore, it could not be viewed as desired on mobile devices. He was aware that page representation was not appropriate to browse on a mobile device. It was also observed that, some existing beliefs of the participants affected their performance. In a browsing task which the participants are supposed to find a support phone number, the majority of the participants expressed the belief that a support number is generally placed at the bottom of the page. Consequently, this belief increased the time for them to complete the browsing tasks.

The items related with system usability consisted of statements to measure simplicity, efficiency, comfort, learnability and productivity of the transcoding system. This experiment did not detect any difference on these aspects between original pages and transcoded pages. A possible explanation for this might be that, transcoded pages offer a new and unfamiliar way of browsing; the participants experienced some mi-

nor difficulties in learning and using the system. One participant commented on the transcoded page that, since it was new, he did not feel that much comfortable. On the other hand, those who perform tasks on original pages were already familiar with the page representation. It seems possible that, after users get used to the way of browsing on transcoded pages, they will be more familiar and hence comfortable using the system.

INFOQUAL part was focused on information representation of the system. The items in this part investigated ease of finding and understanding the required information, as well as the effect and clarity of the information. In general, this study did not find any significant difference between information representations in original and transcoded pages. However, when we analyze the data based on different participant groups, comparing the results of original and transcoded pages showed that, for some of the participant groups, participants who perform tasks on transcoded pages are satisfied more than those who complete the tasks on original pages. These participant groups involve professional participants on web design and development; those who studied web design and development and those who complete bachelor's degree. These results are very important, since they shows that the participants who have experience or educational background on web design and development are more satisfied with the information representation of transcoded pages.

One participant stated that, some of the visual elements do not have a meaningful content and some of these have the same title with other visual elements in the page with meaningful content. In this case, some of the menu items and some content elements in a news page had the same titles, and each of these represented as different elements. The participant indicated that, this may confuse the users since it is not possible to choose which one is the content element, pointing content related labeling or merging the visual blocks. One of the issues emerges from these comments is that, semantic segmentation is as important as the structural segmentation of the page. In this study, we concentrated on structural segmentation and did not take the semantic relations between contents into consideration. This study shows that, a further study with more focus on semantic structure of the page is therefore required. Interestingly, one individual stated that he preferred desktop view on mobile devices, since it enables him to see overall page at once, and another commented that since modern

devices have larger screens, desktop views are more comfortable than mobile mirrors.

The final part of the questionnaire consisted of items related with interface quality of the system. This experiment did not detect any difference on these aspects between original pages and transcoded pages. This part is related with the interface of the client software, therefore, the user interface may be improved with additional functionalities, including gestures to navigate between visual elements.

6.4 Technical Evaluation of Web Page Transcoding Based on Eye Tracking

The main purpose of the web page transcoding application is to improve accessibility of web pages both for disabled people and in mobile devices. Our technical evaluation compared the access time, as well as the resources used such as bandwidth and memory of the device.

6.4.1 Access Time Comparison Results

To investigate the effect of our transcoding implementation for accessibility of disabled people, we compared access time of each visual element in both original and transcoded versions. In this evaluation, we used aDesigner¹ to calculate the time required to access certain visual elements. aDesigner is a disability simulator for designers to check accessibility of their content and applications. It helps to simulate the web environment in low-vision and blind people perspective, so that, designers can see how their web pages are accessed in case of inefficient conditions of voice browsers and screen readers. Using aDesigner in blind people perspective, we calculated the time required to access a particular visual element in six web pages. However, we could not calculate access times for home page of AVG due to technical issues with aDesigner. Access times to visual elements in common scanpaths of the other pages are listed in Table 6.23. The results with * sign indicates that, these are graphical elements and do not provide any textual content or alternative text. Therefore, they are inaccessible for disabled people and aDesigner does not calculate an access time for such elements.

¹ <http://www.eclipse.org/actf/downloads/tools/aDesigner/>, Last Access: 20.07.2014

Table 6.23: Access times to visual elements in common scanpaths

Page	Complexity	Visual Element	Access Time
Apple	Low	1	9 seconds
		2	5 seconds
		3	10 seconds*
Babylon	Low	1	15 seconds*
		2	10 seconds
		3	17 seconds
		4	25 seconds
		5	36 seconds
Yahoo	Medium	1	17 seconds
		2	29 seconds
Godaddy	High	1	53 seconds
		2	51 seconds
		3	66 seconds
BBC	High	1	17 seconds
		2	69 seconds

According to the access time comparison results, we see that time intervals are acceptable in pages with low visual complexity. When the visual complexities of the pages increase, time required to access to the preferred visual elements also increases. For the pages with medium or high visual complexity, some access time results are not reasonable for web browsing. On the other hand, when we transcode the web pages by using our transcoding approach, the transcoding application displays the first visual element in the common scanpath. Therefore, it reduces the time required to access to these visual elements.

6.4.2 Visual Comparison Results

In visual comparison tests, we compared loading time of web pages, the time in which CPU of the device is busy, total number of requests and total size of files downloaded for web pages. In these tests, we first calculated these values by assuming that, client application does not cache previously downloaded CSS and JavaScript files. Then, we repeated the tests by assuming that client application only downloads these files for the first visual element, and does not download again for later requests. In this evaluation, we used visual comparison module of WebPageTest² to calculate the loading time of web pages, CPU used to process the total request sent to the server and total file size downloaded for a web page. We repeated our tests for both original web page and transcoded version with visual elements in the common scanpaths. The

² <http://www.webpagetest.org/>, Last Access: 20.07.2014

Table 6.24: Visual comparison test results

Page	Visual Element	With CSS and JavaScript				Without CSS and JavaScript			
		Load Time (s)	CPU Busy Time (s)	Requests	File Size (KB)	Load Time (s)	CPU Busy Time (s)	Requests	File Size (KB)
Apple	Original	7.148	1.216	39	445.603	-	-	-	-
	VE1	4.309	0.686	19	131.109	-	-	-	-
	VE2	4.071	1.014	18	131.414	2.131	0.374	2	8.106
	VE3	4.158	0.951	18	185.235	1.970	0.234	2	61.894
Babylon	Original	4.890	1.216	42	262.269	-	-	-	-
	VE1	3.433	0.218	9	142.107	-	-	-	-
	VE2	2.391	0.296	7	12.199	1.051	0.702	2	3.993
	VE3	1.970	0.218	7	14.596	0.652	0.109	1	1.214
	VE4	1.976	0.795	7	13.131	0.670	0.436	1	1.249
	VE5	2.001	0.717	7	13.158	0.653	0.358	1	1.195
AVG	Original	8.633	1.029	77	427.53	-	-	-	-
	VE1	4.282	1.263	23	232.277	-	-	-	-
	VE2	4.240	1.310	24	248.006	3.153	0.499	12	52.037
Yahoo	Original	2.728	1.918	50	474.294	-	-	-	-
	VE1	2.393	0.499	23	209.916	-	-	-	-
	VE2	1.098	0.733	16	105.822	1.102	0.265	16	70.697
Godaddy	Original	6.262	1.684	44	417.346	-	-	-	-
	VE1	3.898	0.358	19	179.289	-	-	-	-
	VE2	3.027	1.092	18	135.437	1.932	0.234	3	4.768
	VE3	3.046	0.655	18	138.089	1.329	0.795	3	5.699
BBC	Original	8.613	2.059	70	584.507	-	-	-	-
	VE1	5.742	0.686	25	187.706	-	-	-	-
	VE2	5.651	1.294	31	246.801	2.231	0.858	6	74.978

pages were tested from Dulles, VA in IE 9 environment by using cable connection. Table 6.24 represents the results of these comparisons. As can be seen from this table, transcoding process reduces the number of requests and amount of data transferred, as well as loading time and CPU usage. The performance increases when the client application caches the previously downloaded resources.

6.4.3 Mobile Simulation Results

In mobile environment simulation tests, we calculated average load time and average page size for both original and transcoded web pages. Similar to the visual comparison tests, the results were calculated by assuming that, client application does not cache previously downloaded CSS and JavaScript files. Then, the tests were repeated by assuming that client application only downloads these files for the first visual element, and does not download again. We simulated the mobile environment by using Mobitest module of Akamai³. The tests were conducted in iPhone 5, iOS 6, AT&T

³ <http://mobitest.akamai.com>, Last Access: 20.07.2014

Table 6.25: Mobile simulation tests results

Page	Visual Element	With CSS and JavaScript		Without CSS and JavaScript	
		Avr. Load Time	Avr. Page Size	Avr. Load Time	Avr. Page Size
Apple	Original	4.85s	474.55KB	-	-
	VE1	2.97s	127.24kb	-	-
	VE2	3.11s	127.55kb	1.41s	7.88kb
	VE3	3.75s	180.11kb	1.73s	60.40kb
Babylon	Original	3.72s	259.29kb	-	-
	VE1	2.58s	138.18kb	-	-
	VE2	1.41s	11.36kb	0.99s	3.86kb
	VE3	1.58s	13.70kb	0.70s	1.17kb
	VE4	1.40s	12.27kb	0.69s	1.20kb
AVG	Original	7.20s	401.41kb	-	-
	VE1	4.34s	222.22kb	-	-
	VE2	3.81s	237.95kb	2.03s	47.86kb
Yahoo	Original	2.90s	461.44kb	-	-
	VE1	2.53s	178.31kb	-	-
	VE2	1.48s	101.24kb	1.15s	69.30kb
Godaddy	Original	4.74s	404.79kb	-	-
	VE1	3.22s	171.95kb	-	-
	VE2	2.73s	114.45kb	1.66s	4.19kb
	VE3	2.53s	131.70kb	1.32s	5.09kb
BBC	Original	4.97s	574.68kb	-	-
	VE1	4.08s	170.13kb	-	-
	VE2	3.29s	234.83kb	1.66s	73.26kb

device simulator. Each test was run three times from Cambridge, MA. Table 6.25 represents the results of the mobile environment simulation tests. As can be seen from this table, load time and average page size decreases when the web page is transcoded. Moreover, the performance increases more if the client application caches previously downloaded resources.

6.4.4 Discussion

In this chapter, we presented our technical and user evaluations for our automatic role detection approach and web page transcoding based on eye tracking. The evaluation of visual element identification indicated that, success of middle and higher levels of segmentation were perceived as above average. Moreover, participants preferred higher levels as their best level of segmentation. Furthermore, according to the results of our further investigation, there was a positive correlation between rating and ranking results, meaning that, participants responded in experiments consistently. Finally, the user evaluation for automatic role detection algorithm showed that, our proposed system has around 80% receptive accuracy, but the proposed knowledge base could

be further improved for better accuracy. Our technical evaluation of visual element identification and automatic role detection showed that, even though response time and memory usage when overall web page is processed is high, the results for individual visual elements are reasonable. In a typical application, only a certain level in the web page would be examined; therefore, the system would respond in a practicable time and use feasible amount of memory.

The user evaluation have supported the use of visual element identification and automatic role detection in a transcoding application. Even though the results of transcoded and original pages are very close, participants who performed tasks on transcoded web pages were more satisfied with the system. Moreover, some participant groups which include those who studied web design and development, and professional participants in web design and development gave significantly higher results for transcoded pages. In overall, the results and observations for transcoded web pages were positive and it was demonstrated that web page transcoding based on eye tracking can be used to resolve the web accessibility problems for disabled people and for mobile devices. The technical evaluation of the transcoding implementation also showed that, web page transcoding based on eye tracking decreases the access time to certain visual elements in screen readers. Moreover, in mobile devices, transcoding process reduced CPU usage, average load time and downloaded file size.

According to Alexa, web pages used in this study are very popular and accessed very frequently. However, especially for high complexity pages, it may take unreasonable amount of time to access to a visual element, which a user may be interested most. Transcoding based on eye tracking enables to access these visual elements directly. Therefore, we can conclude that, our transcoding approach has advantages over classical web traveling for disabled people, as well as in mobile web users. Moreover, transcoding approach improved the efficiency in both desktop and mobile platforms.

In conclusion, the analysis presented in this chapter contributes an effective understanding for perceived and technical success of our automatic role detection and web page transcoding approaches. The study also has gone some way towards enhancing our understanding of level preference of participants, which we believe to be useful in many applications.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Web pages include many visual elements to support interaction with users. Web technologies offer a rich set of tools and libraries to create more visually interactive web pages. Unfortunately, far too little attention has been paid to accessibility issues in design process. If these visual elements do not comply with accessibility guidelines, they become inaccessible in alternative platforms, such as when they are accessed with screen readers. Web accessibility investigations show that, majority of web pages have significant web accessibility issues. One way to improve accessibility of such web pages is to transform the representation in alternative forms. There has been many studies to transcode web pages; however, these techniques do not have neither a deep understanding of visual elements nor compatibility with new design trends and web technologies.

In this thesis, we presented our novel approach to automatically detect the roles of visual elements. The system architecture presented in this study has two main components: visual element identification and automatic role detection. In visual element identification, we extended and implemented an existing segmentation algorithm called Vision Based Page Segmentation (VIPS) algorithm to identify each visual element in web page structure. The next component receives these visual elements in a tree structure and detects their corresponding roles. The development process of this component began with construction of a comprehensive role set and knowledge base to describe the characteristics of visual elements. This knowledge is formalized in an ontology. To improve the efficiency of the system, the knowledge in this ontology is converted to rules. Finally, these rules are applied in a probabilistic

fashion to detect the corresponding roles of visual elements in web pages.

To demonstrate the use of our automatic role detection approach, we also presented a transcoding application based on eye tracking as a case study. In this study, we demonstrated transcoding of web pages by using a sequence of visual elements based on eye tracking. Moreover, using an example role path, we allowed user to navigate through the visual elements in web pages. The simulation of this system showed that it is possible to transcode web pages by using eye tracking data as well as underlying roles of visual elements in a web page.

This thesis also presented technical and user evaluations of the proposed approaches and architectures. The technical evaluation has shown reduced access time and memory use in transcoded web pages, that were less than access time and memory use in original web pages. According to the results of evaluation of visual element identification, success of middle and higher levels of segmentation were perceived as above average. Moreover, participants preferred higher level of granularity in web page segmentation. Furthermore, the user evaluation for automatic role detection algorithm has shown that, our proposed system has around 80% receptive accuracy. This accuracy rate can be improved by revising the knowledge base with the feedback retrieved in the experiment. In user evaluation of transcoding, the results of transcoded and original pages were very close; however, transcoded pages had higher ranks in usability tests. Moreover, some participant groups which include those who studied web design and development, and professional participants in web design and development gave significantly higher results for transcoded pages. These results also supported the use of visual element identification and role detection in a transcoding application to improve web accessibility.

Automatic role detection approach presented in this thesis provides a deep understanding of visual elements and their roles. Another significant feature of this approach is the ability to easily modify the knowledge base. Although we analyzed and covered a wide range of heuristic roles and attributes of visual elements in this study, web design trends and web technologies may change in the future. Moreover, different tasks may require a narrowed or extended set of roles as well as different specifications for role attributes. Since proposed system architecture is based on a

modifiable knowledge base, it is possible to adapt it to the changing technologies or web trends. Moreover, application ontologies can also be created for specific tasks.

In conclusion, the research presented in this thesis contributes an effective method for automatically detecting roles of visual elements in web pages by using a set of heuristics. This method is also demonstrated a case study of web page transcoding based on eye tracking. The findings of this research can be used in various fields including information retrieval, web accessibility, intelligent web user interfaces or data mining.

7.1 Future Work

Our investigation on web pages showed that, modern web pages include various types of dynamic visual elements. Examples of these visual elements can be listed as tickers, tabs, slideshows, expansion buttons, auto-suggest lists, calendars, notification boxes and table re-ordering [16]. Typically, these visual elements are invisible at the initial state of web pages. After a certain event is triggered, these visual elements become visible. Moreover, content expansion is also very popular in modern web pages. In this case, content of a visual element is expanded after a certain interaction, such as, clicking on expansion buttons and accordion menus. Identification of dynamic content is also very important for users who are accessing web pages in constraint environments. However, current system architecture does not include a specific mechanism in detecting and understanding dynamic content in web pages. In our future work we are planning to address this by dynamically identifying the new content without completely reprocessing a web page.

Another restriction in this study was that, preliminary results of eMine Scanpath Algorithm provided few web pages and their common scanpaths. Although these results are very promising, generalization of role path requires more comprehensive data. In order to demonstrate role based web page transcoding, we used an example role path in our study. However, it is possible to extend this study by using more extensive set of eye tracking data or different techniques to produce visual element sequences to be used in web page transcoding.

REFERENCES

- [1] Marc Abrams. *World Wide Web: Beyond the Basics*. An Alan R. Apt Book. Prentice Hall PTR, 1998.
- [2] Velibor Adzic, Hari Kalva, and Borko Furht. A survey of multimedia content adaptation for mobile devices. *Multimedia Tools and Applications*, 51:379–396, 2011. 10.1007/s11042-010-0669-x.
- [3] Hamed Ahmadi and Jun Kong. Efficient web browsing on small screens. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 23–30, New York, NY, USA, 2008. ACM.
- [4] Hamed Ahmadi and Jun Kong. User-centric adaptation of web information for small screens. *J. Vis. Lang. Comput.*, 23(1):13–28, February 2012.
- [5] Sadet Alcic and Stefan Conrad. Page segmentation by web content clustering. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, pages 24:1–24:9. ACM, 2011.
- [6] Chieko Asakawa and Hironobu Takagi. Annotation-based transcoding for non-visual web access. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies*, Assets '00, pages 172–179, New York, NY, USA, 2000. ACM.
- [7] Chieko Asakawa and Hironobu Takagi. Transcoding. In Simon Harper and Yeliz Yesilada, editors, *Web Accessibility*, Human-Computer Interaction Series, pages 231–260. Springer London, 2008.
- [8] Shumeet Baluja. 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*, pages 33–42, New York, NY, USA, 2006. ACM.
- [9] Ziv Bar-Yossef and Sridhar Rajagopalan. Template detection via data mining and its applications. In *Proceedings of the 11th international conference on World Wide Web*, WWW '02, pages 580–591, New York, NY, USA, 2002. ACM.
- [10] Armando Barreto. Visual impairments. In Simon Harper and Yeliz Yesilada, editors, *Web Accessibility*, Human-Computer Interaction Series, pages 3–13. Springer London, 2008.

- [11] Eric Bergman and Earl Johnson. Advances in human-computer interaction (vol. 5). chapter Toward Accessible Human-computer Interaction, pages 87–113. Ablex Publishing Corp., Norwood, NJ, USA, 1995.
- [12] Tim Berners-Lee and Mark Fischetti. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Paw Prints, 2008.
- [13] Yevgen Borodin, Jalal Mahmud, I. V. Ramakrishnan, and Amanda Stent. The hearsay non-visual web browser. In *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, W4A '07, pages 128–129, New York, NY, USA, 2007. ACM.
- [14] Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann. Morgan Kaufmann, 2004.
- [15] Dan Brickley and Ramanathan V. Guha. RDF Schema 1.1. <http://www.w3.org/TR/rdf-schema/>, February 2014. Retrieved on 25.06.2014.
- [16] Andy Brown, Caroline Jay, and Simon Harper. Tailored presentation of dynamic web content for audio browsers. *International Journal of Human-Computer Studies*, 70(3):179–196, 2012.
- [17] Radek Burget. Automatic document structure detection for data integration. In *Proceedings of the 10th international conference on Business information systems*, BIS'07, pages 391–397, Berlin, Heidelberg, 2007. Springer-Verlag.
- [18] Radek Burget. Layout based information extraction from html documents. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02*, ICDAR '07, pages 624–628, Washington, DC, USA, 2007. IEEE Computer Society.
- [19] Radek Burget and Ivana Rudolfova. Web page element classification based on visual features. In *Proceedings of the 2009 First Asian Conference on Intelligent Information and Database Systems*, ACIIDS '09, pages 67–72, Washington, DC, USA, 2009. IEEE Computer Society.
- [20] Michael Burmester and Marcus Mast. Repeated web page visits and the scan-path theory: A recurrent pattern detection approach. *Journal of Eye Movement Research*, 3(4):5, 2010.
- [21] Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma, and Ji-Rong Wen. Hierarchical clustering of www image search results using visual, textual and link information. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 952–959, New York, NY, USA, 2004. ACM.
- [22] Deng Cai, Xiaofei He, Ji-Rong Wen, and Wei-Ying Ma. Block-level link analysis. In *Proceedings of the 27th annual international ACM SIGIR conference*

on Research and development in information retrieval, SIGIR '04, pages 440–447, New York, NY, USA, 2004. ACM.

- [23] Deng Cai, Xiaofei He, Wei ying Ma, Ji rong Wen, and Hongjiang Zhang. Organizing www images based on the analysis of page layout and web link structure. In *The 2004 IEEE International Conference on Multimedia and EXPO*, pages 27–30. IEEE, 2004.
- [24] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Extracting content structure for web pages based on visual representation. In *Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications*, AP-Web'03, pages 406–417, Berlin, Heidelberg, 2003. Springer-Verlag.
- [25] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. VIPS: a Vision Based Page Segmentation Algorithm. Technical Report MSR-TR-2003-79, Microsoft Research, 2003.
- [26] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Block-based web search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 456–463, New York, NY, USA, 2004. ACM.
- [27] Jiuxin Cao, Bo Mao, and Junzhou Luo. A segmentation method for web page analysis using shrinking and dividing. *International Journal of Parallel, Emergent and Distributed Systems*, 25(2):93–104, April 2010.
- [28] Anna Cavender and Richard E. Ladner. Hearing impairments. In Simon Harper and Yeliz Yesilada, editors, *Web Accessibility*, Human-Computer Interaction Series, pages 25–35. Springer London, 2008.
- [29] Deepayan Chakrabarti, Ravi Kumar, and Kunal Punera. A graph-theoretic approach to webpage segmentation. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 377–386, New York, NY, USA, 2008. ACM.
- [30] Jinlin Chen, Baoyao Zhou, Jin Shi, Hongjiang Zhang, and Qiu Fengwu. Function-based object model towards website adaptation. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 587–596, New York, NY, USA, 2001. ACM.
- [31] Tianyi Chen, Yeliz Yesilada, and Simon Harper. What input errors do you experience? typing and pointing errors of mobile web users. *International Journal of Human-Computer Studies*, Elsevier, 68:138–157, 2009.
- [32] Yu Chen, Wei-Ying Ma, and Hong-Jiang Zhang. Detecting web page structure for adaptive viewing on small form factor devices. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 225–233, New York, NY, USA, 2003. ACM.

- [33] Yu Chen, Xing Xie, Wei-Ying Ma, and Hong-Jiang Zhang. Adapting web pages for small-screen devices. *IEEE Internet Computing*, 9:50–56, January 2005.
- [34] Wen-Huang Cheng and David Gotz. Context-based page unit recommendation for web-based sensemaking tasks. In *Proceedings of the 14th international conference on Intelligent user interfaces*, IUI '09, pages 107–116, New York, NY, USA, 2009. ACM.
- [35] The World Wide Web Consortium. Facts about W3C. <http://www.w3.org/Consortium/facts#history>, 2012. Retrieved on 09.06.2014.
- [36] The World Wide Web Consortium. OWL 2 web ontology language. <http://www.w3.org/TR/owl2-overview/>, December 2012. Retrieved on 22.05.2014.
- [37] The World Wide Web Consortium. Standards for web applications on mobile: current state and roadmap. <http://www.w3.org/Mobile/mobile-web-app-state/>, April 2014. Retrieved on 09.06.2014.
- [38] James Craig and Michael Cooper. Accessible rich internet applications (WAI-ARIA) 1.0. <http://www.w3.org/TR/wai-aria/>, March 2014. Retrieved on 04.06.2014.
- [39] Suna Durmuş and Kürşat Çağıltay. Kamu kurumu web siteleri ve kullanılabilirlik. In Mehmet Zahid Sobacı and Mete Yıldız, editors, *E-devlet Kamu Yönetimi ve Teknoloji İlişkisinde Güncel Gelişmeler*, pages 293–322. Nobel Yayınevi, 2012.
- [40] Soumitra Dutta and Piero P. Bonissone. Integrating case-based and rule based reasoning: The possibilistic connection. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, pages 281–298, New York, NY, USA, 1991. Elsevier Science Inc.
- [41] Alistair D. N. Edwards. Assistive technologies. In Simon Harper and Yeliz Yesilada, editors, *Web Accessibility*, Human-Computer Interaction Series, pages 142–162. Springer London, 2008.
- [42] Sukru Eraslan, Yeliz Yesilada, and Simon Harper. Correlation between scan-paths and visual elements of web pages. Technical report, Middle East Technical University Northern Cyprus Campus and University of Manchester, 2013.
- [43] Sukru Eraslan, Yeliz Yesilada, and Simon Harper. emine scanpath analysis algorithm. Technical report, Middle East Technical University Northern Cyprus Campus and University of Manchester, 2013.
- [44] Family and Social Policies Ministry. Tracing and control regulation for accessibility. <http://www.resmigazete.gov.tr/eskiler/2013/07/20130720-9.htm>, July 2013. Retrieved on 26.04.2014.

- [45] Ernest Friedman. *Jess in Action: Rule-based Systems in Java*. Manning Publications Co., Greenwich, CT, USA, 2003.
- [46] John H. Gennari, Mark A. Musen, Ray W. Fergerson, William E. Grosso, Monica Crubzy, Henrik Eriksson, Natalya F. Noy, and Samson W. Tu. The evolution of protégé: An environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58:89–123, 2002.
- [47] Joseph H. Goldberg, Mark J. Stimson, Marion Lewenstein, Neil Scott, and Anna M. Wichansky. Eye tracking in web search tasks: Design implications. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, ETRA '02, pages 51–58, New York, NY, USA, 2002. ACM.
- [48] Laura A. Granka, Thorsten Joachims, and Geri Gay. Eye-tracking analysis of user behavior in www search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 478–479, New York, NY, USA, 2004. ACM.
- [49] W3C RDF Working Group. Resource Description Framework (RDF). <http://www.w3.org/RDF/>, February 2014. Retrieved on 25.06.2014.
- [50] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, December 1995.
- [51] Nicola Guarino. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 1st edition, 1998.
- [52] Hui Guo, Jalal Mahmud, Yevgen Borodin, Amanda Stent, and I. V. Ramakrishnan. A general approach for partitioning web page content based on geometric and style information. In *In Proc. of ICDAR 2007*, pages 929–933, 2007.
- [53] Aditya Gupta, Anuj Kumar, Mayank, Vishwa Nath Tripathi, and Sashikala Tapaswi. Mobile web: web manipulation for small displays using multi-level hierarchy page segmentation. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, Mobility '07, pages 599–606, New York, NY, USA, 2007. ACM.
- [54] Vicki L. Hanson, John T. Richards, and Cal Swart. Browser augmentation. In Simon Harper and Yeliz Yesilada, editors, *Web Accessibility*, Human-Computer Interaction Series, pages 215–229. Springer London, 2008.
- [55] Simon Harper, Sean Bechhofer, and Darren Lunn. Taming the inaccessible web. In *Proceedings of the 24th annual ACM international conference on Design of communication*, SIGDOC '06, pages 64–69, New York, NY, USA, 2006. ACM.

- [56] Simon Harper and Yeliz Yesilada. Web authoring for accessibility (WafA). *Web Semant.*, 5(3):175–179, September 2007.
- [57] Simon Harper and Yeliz Yesilada. Web accessibility and guidelines. In Simon Harper and Yeliz Yesilada, editors, *Web Accessibility*, Human-Computer Interaction Series, pages 61–78. Springer London, 2008.
- [58] Gen Hattori, Keiichiro Hoashi, Kazunori Matsumoto, and Fumiaki Sugaya. 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*, pages 361–370, New York, NY, USA, 2007. ACM Press.
- [59] Frederick Hayes-Roth. Rule-based systems. *Commun. ACM*, 28(9):921–932, September 1985.
- [60] Xiaofei He, Deng Cai, Ji-Rong Wen, Wei-Ying Ma, and Hong-Jiang Zhang. Imageseer: Clustering and searching www images using link and page layout analysis. Technical Report MSR-TR-2004-38, Microsoft Technical Report, 2004.
- [61] Philippe Le Hégarret, Ray Whitmer, and Lauren Wood. Document object model (DOM). <http://www.w3.org/DOM/>, January 2005. Retrieved on 09.06.2014.
- [62] Shawn Lawton Henry. Web accessibility initiative (WAI). <http://www.w3.org/WAI/>, March 2011. Retrieved on 04.06.2014.
- [63] Shawn Lawton Henry. Web content accessibility guidelines (WCAG) overview. <http://www.w3.org/WAI/intro/wcag.php>, October 2012. Retrieved on 04.06.2014.
- [64] Shawn Lawton Henry. Authoring tool accessibility guidelines (ATAG) overview. <http://www.w3.org/WAI/intro/atag.php>, October 2013. Retrieved on 04.06.2014.
- [65] Shawn Lawton Henry and Matt May. User agent accessibility guidelines (UAAG) overview. <http://www.w3.org/WAI/intro/uaag.php>, July 2005. Retrieved on 04.06.2014.
- [66] Rinke Hoekstra. *Ontology representation: design patterns and ontologies that make sense*. Frontiers in artificial intelligence and applications. IOS Press, 2009.
- [67] Matthew Horridge and Sean Bechhofer. The owl api: A java api for owl ontologies. *Semant. web*, 2(1):11–21, January 2011.
- [68] Yonghyun Hwang, Jihong Kim, and Eunkyong Seo. Structure-aware web transcoding for mobile devices. *IEEE Internet Computing*, 7(5):14–21, 2003.

- [69] Steven C. Ihde, Paul P. Maglio, Joerg Meyer, and Robert Barrett. Intermediary-based transcoding framework. *IBM Syst. J.*, 40(1):179–192, January 2001.
- [70] Jinbeom Kang, Jaeyoung Yang, and Joongmin Choi. Repetition-based web page segmentation by detecting tag patterns for small-screen devices. *Consumer Electronics, IEEE Transactions on*, 56(2):980–986, May 2010.
- [71] Christian Kohlschütter and Wolfgang Nejdl. A densitometric approach to web page segmentation. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1173–1182. ACM, 2008.
- [72] Milos Kovacevic, Michelangelo Diligenti, Marco Gori, Marco Maggini, and Veljko Milutinovic. Recognition of common areas in a web page using visual information: a possible application in a page classification. In *Proceedings 2002 IEEE International Conference on Data Mining*, pages 250–257, Washington, DC, USA, 2002. IEEE Computer Society.
- [73] Sri H. Kurniawan. Ageing. In Simon Harper and Yeliz Yesilada, editors, *Web Accessibility*, Human-Computer Interaction Series, pages 47–58. Springer London, 2008.
- [74] Serhat Kurt. The accessibility of university web sites: The case of Turkish universities. *Univers. Access Inf. Soc.*, 10(1):101–110, March 2011.
- [75] Clayton Lewis. Cognitive and learning impairments. In Simon Harper and Yeliz Yesilada, editors, *Web Accessibility*, Human-Computer Interaction Series, pages 15–23. Springer London, 2008.
- [76] James R. Lewis. Ibm computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *Int. J. Hum.-Comput. Interact.*, 7(1):57–78, January 1995.
- [77] Antoni Ligêza. Basic structure of rule-based systems. In *Logical Foundations for Rule-Based Systems*, volume 11 of *Studies in Computational Intelligence*, pages 91–96. Springer Berlin Heidelberg, 2006.
- [78] Edward H. Y. Lim, James N. K. Liu, and Raymond S. T. Lee. Computational knowledge and ontology. In *Knowledge Seeker - Ontology Modelling for Information Search and Management*, volume 8 of *Intelligent Systems Reference Library*, pages 3–12. Springer Berlin Heidelberg, 2011.
- [79] Shian-Hua Lin and Jan-Ming Ho. Discovering informative content blocks from web documents. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 588–593, New York, NY, USA, 2002. ACM.
- [80] Bing Liu, Chee Wee Chin, and Hwee Tou Ng. Mining topic-specific concepts and definitions on the web. In *Proceedings of the 12th international conference on World Wide Web*, pages 251–260. ACM, 2003.

- [81] Rui Lopes, Daniel Gomes, and Luís Carriço. Web not for all: A large scale study of web accessibility. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, W4A '10, pages 10:1–10:4, New York, NY, USA, 2010. ACM.
- [82] Darren Lunn, Sean Bechhofer, and Simon Harper. The sadie transcoding platform. In *W4A '08: Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)*, pages 128–129, New York, NY, USA, 2008. ACM.
- [83] Darren Lunn, Sean Bechhofer, and Simon Harper. A user evaluation of the sadie transcoder. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, Assets '08, pages 137–144, New York, NY, USA, 2008. ACM.
- [84] Mehregan Mahdavi, Hamid Khordadi, and Mohammad-Hassan Khoobkar. Web transcoding for mobile devices using a tag-based technique, 2010.
- [85] Jalal Mahmud, Yevgen Borodin, Dipanjan Das, and I. V. Ramakrishnan. Combating information overload in non-visual web access using context. In *Proceedings of the 12th international conference on Intelligent user interfaces*, IUI '07, pages 341–344, New York, NY, USA, 2007. ACM.
- [86] Jalal U. Mahmud, Yevgen Borodin, and I. V. Ramakrishnan. Csurf: a context-driven non-visual web-browser. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 31–40, New York, NY, USA, 2007. ACM.
- [87] Joe Marini. *Document object model : processing structured documents / Joe Marini*. Berkeley, CA: McGraw-Hill/Osborne, c2002., 2002.
- [88] Deborah L. McGuinness, Richard Fikes, James Hendler, and Lynn Andrea Stein. Daml+oil: an ontology language for the semantic web. *Intelligent Systems, IEEE*, 17(5):72–80, Sep 2002.
- [89] Deborah L. McGuinness and Frank van Harmelen. OWL web ontology language. <http://www.w3.org/TR/owl-features/>, February 2004. Retrieved on 22.05.2014.
- [90] Eleni Michailidou. *ViCRAM: Visual Complexity Rankings and Accessibility Metrics*. PhD thesis, 2010.
- [91] Natasa Milic-Frayling and Ralph Sommerer. Smartview: Flexible viewing of web page contents. In *Poster Proceedings of the Eleventh International World Wide Web Conference*, May 2002.
- [92] Tomoyuki Nanno, Suguru Saito, and Manabu Okumura. Structuring web pages based on repetition of elements. *Transactions of Information Processing Society of Japan*, 45(9):2157–2167, 2004.

- [93] Information Technologies Without Barriers Platform of Turkey and The User Experience Professionals Association. Web site accessibility compliance evaluation report. <http://www.engelsizbilisim.org/wp-content/uploads/Eri%C5%9Filebilirlik-Raporu.pdf>, May 2013. Retrieved on 05.07.2014.
- [94] World Health Organization. Visual impairment and blindness. <http://www.who.int/mediacentre/factsheets/fs282/en/>, October 2013. Retrieved on 05.07.2014.
- [95] Heiko Paulheim. *Ontology-based Application Integration [electronic resource] / by Heiko Paulheim*. New York, NY : Springer New York, 2011., 2011.
- [96] Mark Pilgrim. *HTML5: Up and Running*. O'Reilly Media, 2010.
- [97] Alex Poole and Linden J. Ball. Eye tracking in human-computer interaction and usability research: Current status and future. In *Prospects, Chapter in C. Ghaoui (Ed.): Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc, 2005*.
- [98] Dave Raggett, Arnaud Le Hors, and Ian Jacobs. Html 4.01 specification. <http://www.w3.org/TR/html4/>, December 1999. Retrieved on 18.06.2014.
- [99] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, pages 372–422, 1998.
- [100] Uttam Kumar Roy. *Web Technologies*. Oxford higher education. Oxford University Press, Incorporated, 2010.
- [101] Diana Marcela Sánchez, José María Caverro, and Esperanza Marcos Martínez. The road toward ontologies. In Raj Sharman, Rajiv Kishore, and Ram Ramesh, editors, *Ontologies*, volume 14 of *Integrated Series in Information Systems*, pages 3–20. Springer US, 2007.
- [102] Hiroyuki Sano, Shun Shiramatsu, Tadachika Ozono, and Toramatsu Shintani. A web page segmentation method based on page layouts and title blocks. In *International Journal of Computer Science and Network Security*, volume 11, pages 84–90. 2011.
- [103] Richard Schneider and Frank McCown. First steps in archiving the mobile web: Automated discovery of mobile websites. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '13*, pages 53–56, New York, NY, USA, 2013. ACM.
- [104] Robert W. Sebesta. *Programming the World Wide Web / Robert W. Sebesta*. Boston : Pearson/Addison-Wesley, c2005, 2005.

- [105] Rob Shearer, Boris Motik, and Ian Horrocks. Hermit: A highly-efficient owl reasoner. In Catherine Dolbear, Alan Ruttenberg, and Ulrike Sattler, editors, *OWLED*, volume 432 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
- [106] Joel Sklar. *Principles of Web Design: The Web Technologies Series*. HTML Series. Cengage Learning, 2011.
- [107] Ruihua Song, Haifeng Liu, Ji-Rong Wen, and Wei-Ying Ma. Learning block importance models for web pages. In *Proceedings of the 13th international conference on World Wide Web, WWW '04*, pages 203–211, New York, NY, USA, 2004. ACM.
- [108] Hironobu Takagi, Chieko Asakawa, Kentarou Fukuda, and Junji Maeda. Site-wide annotation: Reconstructing existing pages to be accessible. In *Proceedings of the Fifth International ACM Conference on Assistive Technologies, Assets '02*, pages 81–88, New York, NY, USA, 2002. ACM.
- [109] Jim Thatcher. *Web accessibility [electronic resource] : web standards and regulatory compliance / Jim Thatcher ... [et al.]*. Safari Books Online. [Berkeley, Calif.] : Friends of ED ; New York : Distributed by Springer-Verlag, c2006., 2006.
- [110] Alain Tremeau and Nathalie Borel. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30(7):1191 – 1203, 1997.
- [111] Shari Trewin. Physical impairment. In Simon Harper and Yeliz Yesilada, editors, *Web Accessibility*, Human-Computer Interaction Series, pages 37–46. Springer London, 2008.
- [112] Vasudeva Varma. Use of ontologies for organizational knowledge management and knowledge management systems. In Raj Sharman, Rajiv Kishore, and Ram Ramesh, editors, *Ontologies*, volume 14 of *Integrated Series in Information Systems*, pages 21–47. Springer US, 2007.
- [113] Gujjar Vineel. Web page DOM node characterization and its application to page segmentation. In *Proceedings of the 3rd IEEE International Conference on Internet Multimedia Services Architecture and Applications, IMSAA'09*, pages 325–330, Piscataway, NJ, USA, 2009. IEEE Press.
- [114] Xin-Jing Wang, Wei-Ying Ma, Gui-Rong Xue, and Xing Li. Multi-model similarity propagation and its application for web image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia, MULTIMEDIA '04*, pages 944–951, New York, NY, USA, 2004. ACM.
- [115] Yonghyun Whang, Changwoo Jung, Jihong Kim, and Sungkwon Chung. Webalchemist: A web transcoding system for mobile web access in handheld devices. In *Optoelectronic and Wireless Data Management, Processing, Storage, and Retrieval*, pages 102–109, 2001.

- [116] Erik Wilde and David Lowe. *XPath, XLink, XPointer, and XML : a practical guide to Web hyperlinking and transclusion / Erik Wilde, David Lowe*. Boston, MA : Addison-Wesley, c2003., 2003.
- [117] Ou Wu, Yunfei Chen, Bing Li, and Weiming Hu. Evaluating the visual quality of web pages using a computational aesthetic approach. In *Proceedings of the fourth ACM international conference on Web search and data mining, WSDM '11*, pages 337–346, New York, NY, USA, 2011. ACM.
- [118] Peifeng Xiang and Yuanchun Shi. Recovering semantic relations from web pages based on visual cues. In *Proceedings of the 11th international conference on Intelligent user interfaces, IUI '06*, pages 342–344, New York, NY, USA, 2006. ACM.
- [119] Peifeng Xiang, Xin Yang, and Yuanchun Shi. Web page segmentation based on gestalt theory. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 2253–2256, July 2007.
- [120] Xiangye Xiao, Qiong Luo, Dan Hong, and Hongbo Fu. Slicing*-tree based web page transformation for small displays. In *Proceedings of the 14th ACM international conference on Information and knowledge management, CIKM '05*, pages 303–304, New York, NY, USA, 2005. ACM.
- [121] Xiangye Xiao, Qiong Luo, Dan Hong, Hongbo Fu, Xing Xie, and Wei-Ying Ma. Browsing on small displays by transforming web pages into hierarchically structured subpages. *ACM Trans. Web*, 3(1):4:1–4:36, January 2009.
- [122] Yunpeng Xiao, Yang Tao, and Qian Li. Web page adaptation for mobile device. In *Wireless Communications, Networking and Mobile Computing*, 2008.
- [123] Yunpeng Xiao, Yang Tao, and Wenji Li. A dynamic web page adaptation for mobile device based on web2.0. In *Proceedings of the 2008 Advanced Software Engineering and Its Applications*, pages 119–122, Washington, DC, USA, 2008. IEEE Computer Society.
- [124] Xing Xie, Gengxin Miao, Ruihua Song, Ji-Rong Wen, and Wei-Ying Ma. 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*, pages 17–26, Washington, DC, USA, 2005. IEEE Computer Society.
- [125] Xin Yang and Yuanchun Shi. 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, WI-IAT '09*, pages 46–49, Washington, DC, USA, 2009. IEEE Computer Society.

- [126] Yeliz Yesilada, Simon Harper, and Sukru Eraslan. Experiential transcoding: an eyetracking approach. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility, W4A '13*, pages 30:1–30:4, New York, NY, USA, 2013. ACM.
- [127] Yeliz Yesilada, Robert Stevens, Simon Harper, and Carole Goble. Evaluating DANTE: Semantic transcoding for visually disabled users. *ACM Trans. Comput.-Hum. Interact.*, 14(3), 2007.
- [128] Lan Yi, Bing Liu, and Xiaoli Li. Eliminating noisy information in web pages for data mining. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 296–305, New York, NY, USA, 2003. ACM.
- [129] Xinyi Yin and Wee Sun Lee. Using link analysis to improve layout on mobile devices. In *Proceedings of the 13th International Conference on World Wide Web, WWW '04*, pages 338–344, New York, NY, USA, 2004. ACM.
- [130] Xinyi Yin and Wee Sun Lee. 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, WWW '05*, pages 1150–1151, New York, NY, USA, 2005. ACM.
- [131] Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the 12th international conference on World Wide Web*, pages 11–18. ACM, 2003.
- [132] Nicholas C. Zakas. The evolution of web development for mobile devices. *Commun. ACM*, 56(4):42–48, April 2013.

APPENDIX A

HTML AND HTML5 TAG CLASSIFICATION

A.1 HTML Tag Classification

Table A.1: HTML tag classification

Node Type	HTML5 Tags
Invalid Nodes	! . . . , !DOCTYPE, area, base, basefont, col, colgroup, link, map, meta, param, script, style, title
Inline Nodes	a, abbr, acronym, applet, b, bdo, big, button, cite, code, del, dfn, em, font, i, img, input, ins, kbd, label, object, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var
Line Break Nodes	address, blockquote, body, br, caption, center, dd, dir, div, dl, dt, fieldset, form, frame, frameset, h1, head, hr, html, iframe, legend, li, menu, noframes, noscript, ol, optgroup, option, p, pre, table, tbody, td, tfoot, th, thead, tr, ul

A.2 HTML5 Tag Classification

Table A.2: HTML5 tag classification

Node Type	HTML5 Tags
Invalid Nodes	datalist, source
Inline Nodes	audio, bdi, canvas, command, embed, keygen, mark, meter, output, progress, rp, rt, ruby, time, video
Line Break Nodes	article, aside, details, figcaption, figure, footer, header, hgroup, nav, section, summary, wbr

APPENDIX B

MODIFIED RULE SET FOR VISUAL ELEMENT IDENTIFICATION

1. Remove all the invalid nodes. Note that, valid nodes without any child must be kept, since, although they do not have a valid content, they appear in the page layout and might be used as a separator between two different topics. Also, although invalid nodes are not taken into consideration, they may be significant when dynamic content segmentation is handled and criteria for ignoring invisible nodes may be changed in future extensions.
2. If a node has only text node or invalid children, no block extracted.
3. If node has only one child,
 - (a) If child is a text node or virtual text node, then no block extracted.
 - (b) If child is line-break node, then same rules are applied to the child node.
4. If all the children are virtual text nodes of a node, node is put into block pool and we do not divide this rule in next turns.
5. If some of the children has full page width and some of the children are in the same horizontal axis and their total width is equal to the page width, divide this node so that, each full width node is a separate block and the nodes in the same horizontal axis are grouped into one single block.
6. If node is a table and some of its columns have different background color than the others, divide the table into the number separate columns and construct a block for each piece.
7. If one of the child node has bigger font size than its previous siblings, divide node into two blocks. Put the nodes before the child node with bigger font size into the first block, and put the remaining nodes to the second block.

8. If the first child of the node has bigger font size than the remaining children, extract two blocks, one of which is the first child with bigger font size, and the other contains remaining children.
9. If a node contains a child whose tag is HR, BR or any of line-break nodes which has no children, then the node is divided into two as the nodes before the separator and after the separator. For each side of the separator, two new blocks are created and children nodes are put under these blocks. Note that, separator does not extract a block under the main block, it just serves to extract two blocks which other nodes are put into.
10. If a node has some nodes whose line-break spaces are different than each other, divide this node. Each node having larger line-break space will be a separate block, and other nodes next to each other create a block as a whole.
11. If node has at least one child with float value "left" or "right", create three blocks. For each children,
 - (a) If child is left float, put it into the first block.
 - (b) If child is right float, put it into the second block.
 - (c) If child is not both left and right float, put it into the third block. If first block or second block have children, create new blocks for them. Also, create a new block for the child without float.
12. If a node has a child, whose at least one of margin-top and margin-bottom values are nonzero, divide this node into two blocks. Put the sibling nodes before the node with nonzero margin into the first block and put the siblings after the node with nonzero margin into the second block.
 - (a) If child has only nonzero margin-top, put the child into second block.
 - (b) If child has only nonzero margin-bottom, put the child into first block.
 - (c) If child has both nonzero margin-top and nonzero margin-bottom, create a third block and put it between two blocks.
13. If a node has a line-break child containing an image or object, divide node into two blocks. Put the nodes before the child with image or object into the first block, and put the remaining children into the second block.
14. If the first child of the node contains an image or image, put this child as a block, and create a new block for the remaining children.

APPENDIX C

CLASS DEFINITIONS FOR ROLE PROPERTIES

```
Class(ont:tag partial)
Class(ont:position partial)
Class(ont:attribute partial)
Class(ont:style partial)

Class(ont:header partial ont:tag)
Class(ont:nav partial ont:tag)
Class(ont:ul partial ont:tag)
Class(ont:li partial ont:tag)
Class(ont:select partial ont:tag)
Class(ont:input partial ont:tag)
Class(ont:h1 partial ont:tag)
Class(ont:h2 partial ont:tag)
Class(ont:h3 partial ont:tag)
Class(ont:h4 partial ont:tag)
Class(ont:h5 partial ont:tag)
Class(ont:h6 partial ont:tag)
Class(ont:p partial ont:tag)
Class(ont:div partial ont:tag)
Class(ont:object partial ont:tag)
Class(ont:img partial ont:tag)
Class(ont:a partial ont:tag)
Class(ont:iframe partial ont:tag)
Class(ont:caption partial ont:tag)
Class(ont:label partial ont:tag)
Class(ont:link partial ont:tag)
Class(ont:base partial ont:tag)
Class(ont:q partial ont:tag)
Class(ont:cite partial ont:tag)
Class(ont:blockquote partial ont:tag)
Class(ont:footer partial ont:tag)
Class(ont:ol partial ont:tag)
Class(ont:ul partial ont:tag)
Class(ont:form partial ont:tag)
Class(ont:table partial ont:tag)
```

```
Class(ont:hr partial ont:tag)
Class(ont:br partial ont:tag)
Class(ont:strong partial ont:tag)
Class(ont:selection partial ont:tag)
Class(ont:textarea partial ont:tag)
Class(ont:button partial ont:tag)
Class(ont:article partial ont:tag)

Class(ont:top partial ont:position)
Class(ont:bottom partial ont:position)
Class(ont:left partial ont:position)
Class(ont:right partial ont:position)

Class(ont:for partial ont:attribute)
Class(ont:href partial ont:attribute)
Class(ont:hover partial ont:attribute)
Class(ont:command partial ont:attribute)
Class(ont:onsubmit partial ont:attribute)
Class(ont:onclick partial ont:attribute)
Class(ont:onkeyup partial ont:attribute)

Class(ont:display partial ont:style)
Class(ont:border-left partial ont:style)
Class(ont:border-top partial ont:style)
Class(ont:border-bottom partial ont:style)
Class(ont:border-right partial ont:style)
Class(ont:size partial ont:style)
Class(ont:background-image partial ont:style)
Class(ont:font-size partial ont:style)
Class(ont:font-weight partial ont:style)
Class(ont:width partial ont:style)
```

APPENDIX D

OBJECT PROPERTIES

In order to describe the roles, we have used following object property definitions:

- *has_id* specifies the set of id, class, src attribute values which a role may have.
- *has_keyword* specifies the set of keywords which may appear in textual content of a role.
- *must_have_tag* specifies the tags which a role may have in high probability.
- *has_tag* specifies the tags which a role may have in normal probability.
- *may_have_tag* specifies the tags which a role may have in low probability.
- *has_attribute* describes the set of attributes which a role may have, such as onclick, src, and so on.
- *in_position* specifies the position of a role in the page, such as top or bottom.
- *has_child* specifies which children roles may have a role.
- *has_sibling* specifies which sibling roles may have a role.
- *has_parent* specifies which parent roles may have a role.
- *font_size* defines the font size relative to the body font size.
- *font_weight* defines the font weight relative to the body font weight.
- *border* specifies whether the role has borders.
- *font_color* defines if the font color of a role is the same with the body font color.
- *is_composite* specifies whether the structure of a role is composite.
- *has_list_style* defines if the role's list_style property is set.
- *has_background* specifies whether a role has background color or image.
- *has_order* specifies whether a role is in the given order among its siblings.
- *has_size* describes the exact size of a role.
- *word_count* specifies the word count of a role, such as long or short.

- *is_atomic* specifies whether the structure of a role is atomic. It is the inverse of *is_composite* property.
- *relative_size* describes the size of the role as wide or high, rather than numerical value of its sizes.
- *doc* is related with VIPS Algorithm and specifies after which block extraction rule the visual element is constructed.

APPENDIX E

COMMON SCANPATHS

E.1 www.apple.com



Figure E.1: Apple Home Page

E.2 www.avg.com

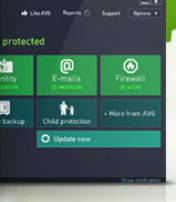

[For Home](#)
[For Business](#)
[For Mobile](#)
[Support](#)





NEW! Ultimate Protection
for your PC and online identity

- Award-winning virus protection
- Faster scan and video streaming
- Protects your identity when shopping and banking





[Protection](#)
[Backup & PC Performance](#)
[Small business solutions](#)



AVG AntiVirus FREE 2013
Essential Protection. Easy to use.



AVG AntiVirus 2013
Advanced Protection. Easy to use.



AVG Premium Security 2013
Ultimate Protection. Privacy. Performance.

About AVG

- Company Profile
- Investor Relations
- License Agreements
- Technology Overview
- Policies
- Contact Us
- Reseller Locator

How can we help

- Antivirus Software
- Business Security
- Mobile Security
- PC Performance
- Free Online Backup
- Free Virus Removal
- Web Threats

Partner with AVG

- Partners & Resellers
- Reseller Center
- AVG CloudCare™
- Affiliates
- Jobs

Media Center

- Press Releases
- AVG Blogs
- Awards
- Customer Reviews
- Resources
- PR Contact

Customer area

- Upgrade to latest version
- Renew my AVG protection
- AVG LiveKive account

Connect with AVG



[United States](#)
[Log into my AVG](#)
[Privacy](#)
[Cookies](#)
[Sitemap](#)

© 2013 AVG Technologies. All rights reserved.

Figure E.2: AVG Home Page

E.3 uk.yahoo.com

Make Y! your homepage

Web Images Video News Shopping More

YAHOO! UK

HOME 19 August 2013 YAHOO! Turkey Yahoo.com SIGN IN New here? Sign up MAIL Check email

Edinburgh Fringe comedy

'We are no longer together but it is fine'

Jigsaw face an awkward relationship question as two of the troupe used to date.
More Fringe in Five

YAHOO! SITES Edit

- Mail
- News **2**
- Sport
- Comedy videos
- Dating
- Lifestyle
- Finance (FTSE 100)
- omg!
- Cars
- Movies
- Shopping
- Games
- Messenger
- Weather (24°C)
- Answers
- Travel
- Horoscopes

More Yahoo! Sites

MY FAVOURITES Edit

- eBay
- Amazon
- Add Favourite

FEATURED PARTNERS

- Pain & Gain
- Alan Partridge

Amazing photo of Rooney's painful isolation

A fan has shared a fabulous picture of the rift between Wayne Rooney and his team-mates. See the pic »

- Man United club page
- Today's papers review
- Seven top flight truths

1 - 4 of 76

Rooney's true isolation Shane Filan opens up Restaurants coin revenge Man City v Newcastle - live!

NEWS SPORT ENTERTAINMENT FINANCE

- Coronation Street Star Fired Over 'Rape' Rap
- Anti-Fracking Protesters Cleared By Police
- Boko Haram Head Abubakar Shekau 'May Be Dead'
- Chinese police 'sacked after nude swim'
- Will HTC One Max have a fingerprint reader?
- Scientists 'have cracked case on Typhoid Mary'
- Ask.fm Overhaul Not Enough, Says Hannah's Dad
- Prince William: George Is A Bit Of A Rascal
- Wallace 'hit man out of the blue'
- Use of UK terror law to detain reporter's partner 'a...

updated 22:07 More: News Videos Weather

Dow: 15,035 -0.31% Nasdaq: 3,600.1 -0.08% FTSE100: 6,465.7 -0.53%

Enter stock symbol Get Quotes

DON'T MISS OUT

Yahoo! Sport

Get the latest football results, news and live coverage.

Trending now

1. Big Brother final
2. Oscar Pistorius
3. Weekend breaks
4. Japan volcano
5. Prince William
6. Egypt unrest
7. Dresses
8. Ellie Goulding
9. House prices
10. Horse trials dea...

giffgaff

For only £10 a month:

- 200 500 minutes
- Unlimited texts
- 1GB Internet
- Free calls to giffgaff

Order a Free SIM

Limited time offer. Terms apply

DAILY OFFERS

- omg! Celebrity gossip to make you gasp!
- Y! Lifestyle From fashion to food to fabulous.

Editor's video picks

Go to video

Why is flooding always so bad in the UK?

- The Lab: Electricity conducting paint
- Milton Jones heckled by Jeremy Clarkson

YAHOO! SITES

- Mail
- Answers
- Careers
- Cars
- Finance
- Flickr
- Games
- Lifestyle
- Mobile
- Travel
- TV
- Yahoo! Ireland
- All Yahoo!
- Fantasy Football
- Latest weather
- Football news
- Career advice
- Facebook Football 2013

FOLLOW YAHOO!

- Follow Us:

YAHOO! FOR YOUR BUSINESS

- Advertise with us

Figure E.3: Yahoo Home Page

E.4 www.godaddy.com

[Log in to My Account](#) or [Create Account](#)

24/7 Support: [\(480\) 505-8877](#) [Support](#)
Hablamos Español

[All Products](#) [Cart](#) [Find your Domain](#) [Build your Website](#) [Get website Hosting](#) [Grow with Web Tools](#) [Search](#)

[Commercials](#) | [Daily Deals](#) | [Bob's Blog](#)

Get started with a domain

2

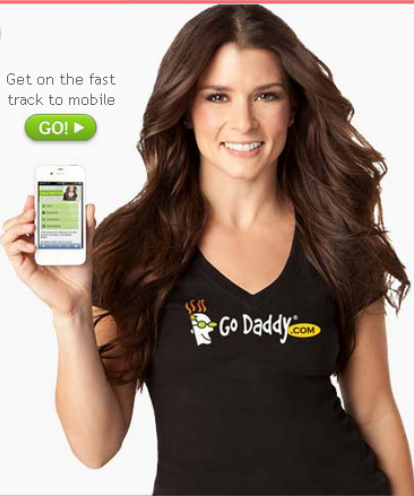
Search for a new domain .com Search

Now just \$9.99 | [FREE .INFO with .COM or .CO Details](#) | [Risk-free guarantee on domain transfers](#)

1

Get on the fast track to mobile

GO!



Go Daddy Website Builder

Fast Track to Mobile

Build your site with Go Daddy and we'll convert it instantly so customers can find you from a desktop, a tablet and now, from a mobile phone. [Buy it »](#)



3

[+1](#) [Tweet](#) [Like](#)

50% off Website Builder.

Plus a FREE domain!
Fast. Easy. Mobile.



[Learn More »](#)

50% off Online Store.

Our best deal ever!
Build a successful online store.



[Learn More »](#)

FREE online bookkeeping.

Simplify your small business.
Keep all your finances in one place.

outright.

[Learn More »](#)

SMALL BUSINESS CENTER

Go Daddy helped take our business from the local fair to the Emmys®.

- MomsOriginals.com

See where we can take you.

[More about domain names](#)

 **BobParsons.me™**
[Visit Bob's Blog »](#)

Bob's Latest: Why successful people determine how much They earn. Never ask for a raise again!
"Excellent Post!" - Shane

*, ** [Click here for product disclaimers and legal policies.](#)

Figure E.4: Godaddy Home Page

BBC News Sport Weather Travel Future TV Radio More... Search

Top News story



William and Kate expecting baby
The Duchess of Cambridge is expecting a baby but is in hospital with morning sickness, St James's Palace announces.

Business Trip



Vancouver: Smart casual
Canada's coastal city leaves many commercial travellers thinking: 'I could really live here.'

Delhi: Extremes and contradictions
Las Vegas: America's conference hub
Chicago: Booming business in the Windy City

News



US fears Syria chemical strike
The United States says it is increasingly concerned Syria's President Bashar al-Assad may resort to using chemical weapons against his people.



Pressure mounts over Israel plans
Britain and France summon Israeli ambassadors in protest at Israel's decision to approve new homes in the West Bank and East Jerusalem.

US town evacuated over explosives
Net freedom fears at UN talks
Dolce and Gabbana in tax trial
Japan orders tunnel inspections
Congoese army returns to Goma

Business

US charges accountants over China
Greece begins buying back bonds

Marketwatch (in delay 15min) Marketwatch Ticker

Dow Jones	12989.89	▼ -35.69	-0.27%
Nasdaq	3009.35	▼ -0.89	-0.03%
FTSE 100	5871.24	▲ 4.42	0.08%
Dax	7435.21	▲ 29.71	0.40%
Cac 40	3586.59	▲ 9.31	0.26%

Sport

England draw Wales and Australia



England draw Australia and Wales in their 2015 Rugby World Cup pool. Scotland face South Africa and Ireland draw France.

Swimming is in a mess, says Adlington
Dettori set for drug test verdict
Wenger irritated by Bould reports
GB's Gandy wants Australia switch

Reaching for the Sky



High-rise heroes
BBC News joins the construction team on the site of London's Leadenhall Building to discover the risks - and thrills - of working at altitude.

Knowledge Economy



Fall on campus
Why the US education has lost its Hollywood ending.

Meet the 'tutor kings and queens'
Qatar's energy wealth fuels ambition for learning
Meet the world's budget English teacher

More from BBC News

Entertainment & Arts



Twilight continues US chart reign
The Twilight finale continues to dominate the North American box office, topping the chart for a third consecutive week.

Health



Partner therapy 'cuts HIV cases'
Treating people with HIV who have uninfected partners significantly reduces transmission rates, researchers in China

Technology



'Revenge porn' site owner threat
Hacktivist collective Anonymous threaten Hunter Moore, the former owner of a "revenge porn" website, as he

Science & Environment



Plastic bulb promises truer light
US researchers say they have developed a new type of lighting that could replace fluorescent bulbs in homes and offices.

Most Popular in News

Shared Read Watched/Listened

Prince William and Kate expecting baby	1
'I woke from coma 14 years in past'	2
'High security' at Duchess hospital	3
Obama lauds Kennedy Center Honorees	4
Man arrested over M1 death crash	5

Figure E.5: BBC Home Page

APPENDIX F

EVALUATION RESULTS OF VISUAL ELEMENT IDENTIFICATION

F.1 Rating Results With Respect to User Profiles

Table F.1: Rating results for pages with 3 levels with respect to user profiles

Participant Profile		Level 1			Level 2			Level 3		
		Mean	N	Std. Dev.	Mean	N	Std. Dev.	Mean	N	Std. Dev.
Age	18-24	2.33	3	0.58	3.33	3	1.15	4.00	3	1.73
	35-54	2.00	5	1.00	3.40	5	1.14	3.80	5	1.10
Education	High/Secondary School	2.00	2	0.00	3.00	2	1.41	3.50	2	2.12
	Associate's Deg.	1.00	1	0.00	3.00	1	0.00	3.00	1	0.00
	Bachelor's Deg.	3.00	2	0.00	4.00	2	0.00	5.00	2	0.00
	Master's Deg.	2.50	2	0.71	4.00	2	1.41	4.00	2	1.41
	Doctorate	1.00	1	0.00	2.00	1	0.00	3.00	1	0.00
Gender	Female	2.00	5	1.00	3.20	5	0.84	3.80	5	1.10
	Male	2.33	3	0.58	3.67	3	1.53	4.00	3	1.73
Level of Expertise	Professional	2.67	3	0.58	4.00	3	1.00	4.33	3	1.15
	Intermediate	2.00	3	1.00	3.33	3	1.15	4.33	3	1.15
	Novice/Beginner	1.50	2	0.71	2.50	2	0.71	2.50	2	0.71
Current Status	Worked	2.17	6	0.98	3.50	6	1.05	4.00	6	1.10
	Studied	2.00	1	0.00	4.00	1	0.00	5.00	1	0.00
	Other	2.00	1	0.00	2.00	1	0.00	2.00	1	0.00

Table F.2: Rating results for pages with 4 levels with respect to user profiles

Participant Profile		Level 1			Level 2			Level 3			Level 4		
		Mean	N	Std. Dev.	Mean	N	Std. Dev.	Mean	N	Std. Dev.	Mean	N	Std. Dev.
Age	18-24	2.75	4	1.71	2.50	4	1.00	3.00	4	1.41	3.75	4	0.50
	25-34	2.75	8	1.58	3.00	8	0.76	3.75	8	1.28	3.38	8	0.92
	35-54	3.78	9	1.48	3.78	9	1.20	4.00	9	1.32	4.00	9	1.32
	55+	3.50	2	0.71	3.00	2	0.00	2.50	2	0.71	2.50	2	0.71
Education	High/Secondary School	3.33	3	1.53	2.67	3	1.15	3.00	3	1.73	3.67	3	0.58
	Associate's Degree	3.00	1	0.00	4.00	1	0.00	4.00	1	0.00	5.00	1	0.00
	Bachelor's Degree	2.50	6	1.52	3.17	6	1.17	3.67	6	0.82	4.33	6	0.82
	Master's Degree	2.67	6	1.63	3.00	6	0.89	3.83	6	1.47	3.17	6	0.75
Gender	Doctorate	4.00	7	1.15	3.57	7	1.13	3.57	7	1.62	3.43	7	1.51
	Female	2.64	11	1.29	2.82	11	0.75	3.09	11	1.04	3.64	11	0.92
	Male	3.75	12	1.54	3.58	12	1.16	4.08	12	1.38	3.58	12	1.24
	Professional	3.00	8	1.69	3.25	8	1.28	3.75	8	1.58	3.50	8	1.41
Level of Expertise	Intermediate	3.42	12	1.38	3.25	12	0.97	3.50	12	1.17	3.75	12	0.97
	Novice/Beginner	3.00	3	2.00	3.00	3	1.00	3.67	3	1.53	3.33	3	0.58
	Worked	2.86	14	1.51	3.21	14	1.19	3.64	14	1.34	3.79	14	1.25
	Studied	4.33	3	1.15	3.33	3	1.15	3.67	3	1.53	3.67	3	0.58
Current Status	Hobby	3.50	6	1.52	3.17	6	0.75	3.50	6	1.38	3.17	6	0.75

Table F.3: Rating results for pages with 5 and more levels with respect to user profiles

Participant Profile		Level 1			Level 2			Level 3			Level 4			Level 5		
		Mean	N	Std. Dev.	Mean	N	Std. Dev.	Mean	N	Std. Dev.	Mean	N	Std. Dev.	Mean	N	Std. Dev.
Age	18-24	2.14	35	1.54	2.46	35	1.24	3.14	35	1.09	3.54	35	1.20	3.86	35	1.24
	25-34	2.25	75	1.25	2.84	75	1.24	3.35	75	0.99	3.36	75	1.01	3.59	75	1.15
	35-54	2.53	51	1.06	2.88	51	0.91	3.29	51	0.90	3.78	51	0.88	3.76	51	1.16
	55+	2.40	10	0.70	3.20	10	0.63	3.50	10	0.71	3.10	10	1.10	2.60	10	0.70
Education	High/Secondary Sch.	2.84	19	1.71	3.00	19	1.29	3.47	19	1.07	3.95	19	1.08	3.84	19	1.34
	Associate's Degree	2.60	10	1.35	2.50	10	0.97	3.00	10	0.67	3.60	10	0.70	3.10	10	0.99
	Bachelor's Degree	1.67	36	0.89	2.33	36	0.96	3.19	36	0.92	3.72	36	1.03	4.28	36	0.97
	Master's Degree	2.37	57	1.17	2.89	57	1.22	3.16	57	1.10	3.12	57	1.12	3.16	57	1.19
Gender	Doctorate	2.49	49	1.16	3.00	49	1.04	3.53	49	0.82	3.61	49	0.84	3.76	49	1.01
	Female	2.11	74	1.20	2.64	74	1.01	3.19	74	0.87	3.59	74	0.96	3.72	74	1.15
	Male	2.48	97	1.25	2.92	97	1.20	3.38	97	1.04	3.44	97	1.08	3.58	97	1.20
	Professional	2.20	71	1.21	2.68	71	1.22	3.11	71	1.14	3.14	71	1.19	3.31	71	1.30
Level of Expertise	Intermediate	2.47	77	1.23	2.86	77	0.97	3.45	77	0.79	3.82	77	0.85	3.86	77	1.10
	Novice/Beginner	2.22	23	1.35	2.96	23	1.36	3.35	23	0.88	3.61	23	0.66	3.91	23	0.73
	Worked	2.08	124	1.10	2.63	124	1.06	3.25	124	1.02	3.50	124	1.12	3.66	124	1.26
	Studied	3.62	21	1.16	3.33	21	1.20	3.48	21	0.87	3.76	21	0.83	3.57	21	1.03
Current Status	Hobby	2.54	24	1.28	3.25	24	1.22	3.46	24	0.78	3.33	24	0.70	3.50	24	0.88
	Other	1.00	2	0.00	2.00	2	1.41	2.50	2	0.71	3.50	2	0.71	4.50	2	0.71

F.2 Correlation Between Rating Results and Participant Profiles

Table F.4: Pearson's Chi Square Test results for participant groups

	Level 1	Level 2	Level 3	Level 4	Level 5
Age	$X^2(12, N=202)=22.846,$ p=0.029	$X^2(12, N=202)=26.924,$ p=0.008	$X^2(12, N=202)=8.868,$ p=0.714	$X^2(15, N=202)=29.379,$ p=0.014	$X^2(15, N=202)=31.187,$ p=0.008
Education	$X^2(16, N=202)=34.291,$ p=0.005	$X^2(16, N=202)=15.224,$ p=0.508	$X^2(16, N=202)=23.734,$ p=0.095	$X^2(20, N=202)=34.527,$ p=0.023	$X^2(20, N=202)=41.885,$ p=0.003
Gender	$X^2(4, N=202)=7.618,$ p=0.107	$X^2(4, N=202)=10.343,$ p=0.035	$X^2(4, N=202)=10.677,$ p=0.03	$X^2(5, N=202)=4.186,$ p=0.523	$X^2(5, N=202)=2.774,$ p=0.735
Level	$X^2(8, N=202)=9.507,$ p=0.301	$X^2(8, N=202)=13.492,$ p=0.096	$X^2(8, N=202)=23.693,$ p=0.003	$X^2(10, N=202)=31.698,$ p=0	$X^2(10, N=202)=22.441,$ p=0.013
Status	$X^2(12, N=202)=38.231,$ p=0	$X^2(12, N=202)=17.747,$ p=0.124	$X^2(12, N=202)=10.165,$ p=0.601	$X^2(15, N=202)=27.267,$ p=0.027	$X^2(15, N=202)=16.647,$ p=0.34

F.3 Ranking Results With Respect to User Profiles

Table F.5: Ranking results for pages with 3 levels, with respect to user profiles

			Level 1	Level 2	Level 3
Age	18-24	Best Level	1 (33.3%)	1 (33.3%)	1 (33.3%)
		2nd Level	0 (0%)	1 (33.3%)	2 (66.7%)
		Worst Level	2 (66.7%)	1 (33.3%)	0 (0%)
	35-54	Best Level	0 (0%)	2 (40%)	3 (60%)
		2nd Level	0 (0%)	3 (60%)	2 (40%)
		Worst Level	5 (100%)	0 (0%)	0 (0%)
Gender	Female	Best Level	0 (0%)	1 (20%)	4 (80%)
		2nd Level	0 (0%)	4 (80%)	1 (20%)
		Worst Level	5 (100%)	0 (0%)	0 (0%)
	Male	Best Level	1 (33.3%)	2 (66.7%)	0 (0%)
		2nd Level	0 (0%)	0 (0%)	3 (100%)
		Worst Level	2 (66.7%)	1 (33.3%)	0 (0%)
Education	High/Secondary Sch.	Best Level	0 (0%)	1 (50%)	1 (50%)
		2nd Level	0 (0%)	1 (50%)	1 (50%)
		Worst Level	2 (100%)	0 (0%)	0 (0%)
	Bachelor's Degree	Best Level	0 (0%)	1 (100%)	0 (0%)
		2nd Level	0 (0%)	0 (0%)	1 (100%)
		Worst Level	1 (100%)	0 (0%)	0 (0%)
	Master's Degree	Best Level	1 (50%)	0 (0%)	1 (50%)
		2nd Level	0 (0%)	1 (50%)	1 (50%)
		Worst Level	1 (50%)	1 (50%)	0 (0%)
	Associate's Degree	Best Level	0 (0%)	1 (50%)	1 (50%)
		2nd Level	0 (0%)	1 (50%)	1 (50%)
		Worst Level	2 (100%)	0 (0%)	0 (0%)
	Doctorate	Best Level	0 (0%)	0 (0%)	1 (100%)
		2nd Level	0 (0%)	1 (100%)	0 (0%)
		Worst Level	1 (100%)	0 (0%)	0 (0%)
Level of Expertise	Professional	Best Level	1 (33.3%)	1 (33.3%)	1 (33.3%)
		2nd Level	0 (0%)	1 (33.3%)	2 (66.7%)
		Worst Level	2 (66.7%)	1 (33.3%)	0 (0%)
	Intermediate	Best Level	0 (0%)	0 (0%)	3 (100%)
		2nd Level	0 (0%)	3 (100%)	0 (0%)
		Worst Level	3 (100%)	0 (0%)	0 (0%)
	Novice/Beginner	Best Level	0 (0%)	2 (100%)	0 (0%)
		2nd Level	0 (0%)	0 (0%)	2 (100%)
		Worst Level	2 (100%)	0 (0%)	0 (0%)

Table F.6: Ranking results for pages with 3 levels, with respect to user profiles (cont.)

			Level 1	Level 2	Level 3
Current Status	Worked	Best Level	1 (16.7%)	2 (33.3%)	3 (50%)
		2nd Level	0 (0%)	3 (50%)	3 (50%)
		Worst Level	5 (83.3%)	1 (16.7%)	0 (0%)
	Studied	Best Level	0 (0%)	0 (0%)	1 (100%)
		2nd Level	0 (0%)	1 (100%)	0 (0%)
		Worst Level	1 (100%)	0 (0%)	0 (0%)
	Other	Best Level	0 (0%)	1 (100%)	0 (0%)
		2nd Level	0 (0%)	0 (0%)	1 (100%)
		Worst Level	1 (100%)	0 (0%)	0 (0%)

Table F.7: Ranking results for pages with 4 levels, with respect to user profiles (Part 1: Age and gender)

			Level 1	Level 2	Level 3	Level 4
Age	"18-24"	Best Level	0 (0.0%)	0 (0.0%)	1 (25.0%)	3 (75.0%)
		2nd Level	2 (50.0%)	0 (0.0%)	1 (25.0%)	1 (25.0%)
		3rd Level	1 (25.0%)	3 (75.0%)	0 (0.0%)	0 (0.0%)
		Worst Level	1 (25.0%)	1 (25.0%)	2 (50.0%)	0 (0.0%)
	"25-34"	Best Level	2 (25.0%)	0 (0.0%)	2 (25.0%)	4 (50.0%)
		2nd Level	0 (0.0%)	1 (12.5%)	6 (75.0%)	1 (12.5%)
		3rd Level	0 (0.0%)	7 (87.5%)	0 (0.0%)	1 (12.5%)
		Worst Level	6 (75.0%)	0 (0.0%)	0 (0.0%)	2 (25.0%)
	"35-54"	Best Level	3 (33.3%)	0 (0.0%)	1 (11.1%)	5 (55.6%)
		2nd Level	0 (0.0%)	3 (33.3%)	5 (55.6%)	1 (11.1%)
		3rd Level	0 (0.0%)	6 (66.7%)	3 (33.3%)	0 (0.0%)
		Worst Level	6 (66.7%)	0 (0.0%)	0 (0.0%)	3 (33.3%)
	"55+"	Best Level	1 (50.0%)	0 (0.0%)	1 (50.0%)	0 (0.0%)
		2nd Level	0 (0.0%)	2 (100.0%)	0 (0.0%)	0 (0.0%)
		3rd Level	1 (50.0%)	0 (0.0%)	1 (50.0%)	0 (0.0%)
		Worst Level	0 (0.0%)	0 (0.0%)	0 (0.0%)	2 (100.0%)
Gender	Female	Best Level	1 (9.1%)	0 (0.0%)	3 (27.3%)	7 (63.6%)
		2nd Level	1 (9.1%)	2 (18.2%)	6 (54.5%)	2 (18.2%)
		3rd Level	2 (18.2%)	8 (72.7%)	1 (9.1%)	0 (0.0%)
		Worst Level	7 (63.6%)	1 (9.1%)	1 (9.1%)	2 (18.2%)
	Male	Best Level	5 (41.7%)	0 (0.0%)	2 (16.7%)	5 (41.7%)
		2nd Level	1 (8.3%)	4 (33.3%)	6 (50.0%)	1 (8.3%)
		3rd Level	0 (0.0%)	8 (66.7%)	3 (25.0%)	1 (8.3%)
		Worst Level	6 (50.0%)	0 (0.0%)	1 (8.3%)	5 (41.7%)

Table F.8: Ranking results for pages with 4 levels, with respect to user profiles
(Part 2: Education, level of expertise and current status in web design and development)

			Level 1	Level 2	Level 3	Level 4
Education	High/Secondary School	Best Level	0 (0.0%)	0 (0.0%)	1 (33.3%)	2 (66.7%)
		2nd Level	2 (66.7%)	0 (0.0%)	0 (0.0%)	1 (33.3%)
		3rd Level	1 (33.3%)	2 (66.7%)	0 (0.0%)	0 (0.0%)
		Worst Level	0 (0.0%)	1 (33.3%)	2 (66.7%)	0 (0.0%)
	Bachelor's Degree	Best Level	1 (100.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
		2nd Level	0 (0.0%)	1 (100.0%)	0 (0.0%)	0 (0.0%)
		3rd Level	0 (0.0%)	0 (0.0%)	1 (100.0%)	0 (0.0%)
		Worst Level	0 (0.0%)	0 (0.0%)	0 (0.0%)	1 (100.0%)
	Master's Degree	Best Level	0 (0.0%)	0 (0.0%)	0 (0.0%)	6 (100.0%)
		2nd Level	0 (0.0%)	0 (0.0%)	6 (100.0%)	0 (0.0%)
		3rd Level	0 (0.0%)	6 (100.0%)	0 (0.0%)	0 (0.0%)
		Worst Level	6 (100.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Associate's Degree	Best Level	1 (16.7%)	0 (0.0%)	3 (50.0%)	2 (33.3%)
		2nd Level	0 (0.0%)	1 (16.7%)	3 (50.0%)	2 (33.3%)
		3rd Level	0 (0.0%)	5 (83.3%)	0 (0.0%)	1 (16.7%)
		Worst Level	5 (83.3%)	0 (0.0%)	0 (0.0%)	1 (16.7%)
	Doctorate	Best Level	4 (57.1%)	0 (0.0%)	1 (14.3%)	2 (28.6%)
		2nd Level	0 (0.0%)	4 (57.1%)	3 (42.9%)	0 (0.0%)
		3rd Level	1 (14.3%)	3 (42.9%)	3 (42.9%)	0 (0.0%)
		Worst Level	2 (28.6%)	0 (0.0%)	0 (0.0%)	5 (71.4%)
Level of Expertise	Professional	Best Level	2 (25.0%)	0 (0.0%)	2 (25.0%)	4 (50.0%)
		2nd Level	0 (0.0%)	3 (37.5%)	4 (50.0%)	1 (12.5%)
		3rd Level	0 (0.0%)	5 (62.5%)	2 (25.0%)	1 (12.5%)
		Worst Level	6 (75.0%)	0 (0.0%)	0 (0.0%)	2 (25.0%)
	Intermediate	Best Level	3 (25.0%)	0 (0.0%)	2 (16.7%)	7 (58.3%)
		2nd Level	2 (16.7%)	3 (25.0%)	6 (50.0%)	1 (8.3%)
		3rd Level	2 (16.7%)	8 (66.7%)	2 (16.7%)	0 (0.0%)
		Worst Level	5 (41.7%)	1 (8.3%)	2 (16.7%)	4 (33.3%)
	Novice/Beginner	Best Level	1 (33.3%)	0 (0.0%)	1 (33.3%)	1 (33.3%)
		2nd Level	0 (0.0%)	0 (0.0%)	2 (66.7%)	1 (33.3%)
		3rd Level	0 (0.0%)	3 (100.0%)	0 (0.0%)	0 (0.0%)
		Worst Level	2 (66.7%)	0 (0.0%)	0 (0.0%)	1 (33.3%)
Current Status	Worked	Best Level	2 (14.3%)	0 (0.0%)	2 (14.3%)	10 (71.4%)
		2nd Level	1 (7.1%)	3 (21.4%)	9 (64.3%)	1 (7.1%)
		3rd Level	0 (0.0%)	11 (78.6%)	2 (14.3%)	1 (7.1%)
		Worst Level	11 (78.6%)	0 (0.0%)	1 (7.1%)	2 (14.3%)
	Studied	Best Level	1 (33.3%)	0 (0.0%)	1 (33.3%)	1 (33.3%)
		2nd Level	1 (33.3%)	1 (33.3%)	0 (0.0%)	1 (33.3%)
		3rd Level	1 (33.3%)	1 (33.3%)	1 (33.3%)	0 (0.0%)
		Worst Level	0 (0.0%)	1 (33.3%)	1 (33.3%)	1 (33.3%)
	Hobby	Best Level	3 (50.0%)	0 (0.0%)	2 (33.3%)	1 (16.7%)
		2nd Level	0 (0.0%)	2 (33.3%)	3 (50.0%)	1 (16.7%)
		3rd Level	1 (16.7%)	4 (66.7%)	1 (16.7%)	0 (0.0%)
		Worst Level	2 (33.3%)	0 (0.0%)	0 (0.0%)	4 (66.7%)

Table F.9: Ranking results for pages with 5 and more levels, with respect to user profiles

(Part 1: Age, gender and level of expertise)

			Level 1	Level 2	Level 3	Level 4	Level 5
Age	18-24	Best Level	2 (5.7)	4 (11.4)	1 (2.9)	3 (8.6)	25 (71.4)
		2nd Level	4 (11.4)	2 (5.7)	3 (8.6)	24 (68.6)	2 (5.7)
		3rd Level	1 (2.9)	2 (5.7)	27 (77.1)	4 (11.4)	1 (2.9)
		4th Level	1 (2.9)	26 (74.3)	4 (11.4)	2 (5.7)	2 (5.7)
		Worst Level	27 (77.1)	1 (2.9)	0 (0)	2 (5.7)	5 (14.3)
	25-34	Best Level	4 (5.3)	15 (20.0)	12 (16.0)	13 (17.3)	31 (41.3)
		2nd Level	8 (10.7)	9 (12.0)	10 (13.3)	36 (48.0)	12 (16.0)
		3rd Level	7 (9.3)	1 (1.3)	51 (68.0)	10 (13.3)	6 (8.0)
		4th Level	7 (9.3)	45 (60.0)	2 (2.7)	15 (20.0)	6 (8.0)
		Worst Level	49 (65.3)	5 (6.7)	0 (0)	1 (1.3)	20 (26.7)
	35-54	Best Level	6 (11.8)	2 (3.9)	5 (9.8)	13 (25.5)	25 (49.0)
		2nd Level	1 (2.0)	4 (7.8)	6 (11.8)	27 (52.9)	13 (25.5)
		3rd Level	5 (9.8)	9 (17.6)	32 (62.7)	3 (5.9)	2 (3.9)
		4th Level	5 (9.8)	33 (64.7)	4 (7.8)	5 (9.8)	4 (7.8)
		Worst Level	34 (66.7)	3 (5.9)	4 (7.8)	3 (5.9)	7 (13.7)
	55+	Best Level	0 (0)	2 (20.0)	4 (40.0)	2 (20.0)	2 (20.0)
		2nd Level	1 (10.0)	0 (0)	4 (40.0)	5 (50.0)	0 (0)
		3rd Level	5 (50.0)	4 (40.0)	2 (20.0)	2 (20.0)	2 (20.0)
		4th Level	0 (0)	4 (40.0)	0 (0)	0 (0)	1 (10.0)
		Worst Level	4 (40.0)	0 (0)	0 (0)	1 (10.0)	5 (50.0)
Gender	Female	Best Lev.	3 (4.1)	9 (12.2)	9 (12.2)	15 (20.3)	38 (51.4)
		2nd Lev.	7 (9.5)	6 (8.1)	6 (8.1)	42 (56.8)	13 (17.6)
		3rd Lev.	3 (4.1)	5 (6.8)	55 (74.3)	7 (9.5)	4 (5.4)
		4th Lev.	6 (8.1)	53 (71.6)	4 (5.4)	8 (10.8)	3 (4.1)
		Worst Lev.	55 (74.3)	1 (1.4)	0 (0)	2 (2.7)	16 (21.6)
	Male	Best Lev.	9 (9.3)	14 (14.4)	13 (13.4)	16 (16.5)	45 (46.4)
		2nd Lev.	7 (7.2)	9 (9.3)	17 (17.5)	50 (51.5)	14 (14.4)
		3rd Lev.	10 (10.3)	11 (11.3)	57 (58.8)	12 (12.4)	7 (7.2)
		4th Lev.	12 (12.4)	55 (56.7)	6 (6.2)	14 (14.4)	10 (10.3)
		Worst Lev.	59 (60.8)	8 (8.2)	4 (4.1)	5 (5.2)	21 (21.6)
Level of Expertise	Professional	Best Level	4 (5.6)	9 (12.7)	13 (18.3)	10 (14.1)	35 (49.3)
		2nd Level	7 (9.9)	9 (12.7)	6 (8.5)	38 (53.5)	11 (15.5)
		3rd Level	7 (9.9)	5 (7.0)	46 (64.8)	8 (11.3)	5 (7.0)
		4th Level	8 (11.3)	46 (64.8)	3 (4.2)	12 (16.9)	2 (2.8)
		Worst Level	45 (63.4)	2 (2.8)	3 (4.2)	3 (4.2)	18 (25.4)
	Intermediate	Best Level	6 (7.8)	8 (10.4)	8 (10.4)	17 (22.1)	38 (49.4)
		2nd Level	5 (6.5)	5 (6.5)	11 (14.3)	43 (55.8)	13 (16.9)
		3rd Level	4 (5.2)	10 (13.0)	51 (66.2)	7 (9.1)	5 (6.5)
		4th Level	9 (11.7)	49 (63.6)	6 (7.8)	6 (7.8)	7 (9.1)
		Worst Level	53 (68.8)	5 (6.5)	1 (1.3)	4 (5.2)	14 (18.2)
	Novice/ Beginner	Best Level	2 (8.7)	6 (26.1)	1 (4.3)	4 (17.4)	10 (43.5)
		2nd Level	2 (8.7)	1 (4.3)	6 (26.1)	11 (47.8)	3 (13.0)
		3rd Level	2 (8.7)	1 (4.3)	15 (65.2)	4 (17.4)	1 (4.3)
		4th Level	1 (4.3)	13 (56.5)	1 (4.3)	4 (17.4)	4 (17.4)
		Worst Level	16 (69.6)	2 (8.7)	0 (0)	0 (0)	5 (21.7)

Table F.10: Ranking results for pages with 5 and more levels, with respect to user profiles

(Part 2: Education and current status in web design and development)

			Level 1	Level 2	Level 3	Level 4	Level 5
Education	High/ Secondary School	Best Level	2 (10.5)	4 (21.1)	1 (5.3)	3 (15.8)	9 (47.4)
		2nd Level	4 (21.1)	2 (10.5)	3 (15.8)	8 (42.1)	2 (10.5)
		3rd Level	1 (5.3)	2 (10.5)	11 (57.9)	4 (21.1)	1 (5.3)
		4th Level	1 (5.3)	10 (52.6)	4 (21.1)	2 (10.5)	2 (10.5)
		Worst Level	11 (57.9)	1 (5.3)	0 (0)	2 (10.5)	5 (26.3)
	Bachelor's Degree	Best Level	0 (0)	0 (0)	1 (2.8)	2 (5.6)	33 (91.7)
		2nd Level	0 (0)	0 (0)	0 (0)	34 (94.4)	2 (5.6)
		3rd Level	0 (0)	1 (2.8)	35 (97.2)	0 (0)	0 (0)
		4th Level	0 (0)	35 (97.2)	0 (0)	0 (0)	1 (2.8)
		Worst Level	36 (100.0)	0 (0)	0 (0)	0 (0)	0 (0)
	Master's Degree	Best Level	5 (8.8)	12 (21.1)	10 (17.5)	11 (19.3)	19 (33.3)
		2nd Level	9 (15.8)	8 (14.0)	8 (14.0)	22 (38.6)	10 (17.5)
		3rd Level	6 (10.5)	4 (7.0)	34 (59.6)	9 (15.8)	4 (7.0)
		4th Level	5 (8.8)	30 (52.6)	4 (7.0)	14 (24.6)	4 (7.0)
		Worst Level	32 (56.1)	3 (5.3)	1 (1.8)	1 (1.8)	20 (35.1)
	Associate's Degree	Best Level	2 (20.0)	0 (0)	1 (10.0)	5 (50.0)	2 (20.0)
		2nd Level	0 (0)	2 (20.0)	4 (40.0)	2 (20.0)	2 (20.0)
		3rd Level	1 (10.0)	4 (40.0)	5 (50.0)	0 (0)	0 (0)
		4th Level	1 (10.0)	4 (40.0)	0 (0)	2 (20.0)	3 (30.0)
		Worst Level	6 (60.0)	0 (0)	0 (0)	1 (10.0)	3 (30.0)
	Doctorate	Best Level	3 (6.1)	7 (14.3)	9 (18.4)	10 (20.4)	20 (40.8)
		2nd Level	1 (2.0)	3 (6.1)	8 (16.3)	26 (53.1)	11 (22.4)
		3rd Level	5 (10.2)	5 (10.2)	27 (55.1)	6 (12.2)	6 (12.2)
		4th Level	11 (22.4)	29 (59.2)	2 (4.1)	4 (8.2)	3 (6.1)
		Worst Level	29 (59.2)	5 (10.2)	3 (6.1)	3 (6.1)	9 (18.4)
Current Status	Worked	Best Level	5 (4.0)	10 (8.1)	16 (12.9)	22 (17.7)	71 (57.3)
		2nd Level	8 (6.5)	9 (7.3)	11 (8.9)	76 (61.3)	20 (16.1)
		3rd Level	8 (6.5)	10 (8.1)	88 (71.0)	10 (8.1)	8 (6.5)
		4th Level	12 (9.7)	89 (71.8)	5 (4.0)	13 (10.5)	5 (4.0)
		Worst Level	91 (73.4)	6 (4.8)	4 (3.2)	3 (2.4)	20 (16.1)
	Studied	Best Level	4 (19.0)	6 (28.6)	3 (14.3)	5 (23.8)	3 (14.3)
		2nd Level	6 (28.6)	5 (23.8)	3 (14.3)	3 (14.3)	4 (19.0)
		3rd Level	3 (14.3)	3 (14.3)	11 (52.4)	3 (14.3)	1 (4.8)
		4th Level	1 (4.8)	6 (28.6)	4 (19.0)	7 (33.3)	3 (14.3)
		Worst Level	7 (33.3)	1 (4.8)	0 (0)	3 (14.3)	10 (47.6)
	Hobby	Best Level	3 (12.5)	7 (29.2)	3 (12.5)	4 (16.7)	7 (29.2)
		2nd Level	2 (8.3)	1 (4.2)	9 (37.5)	11 (45.8)	3 (12.5)
		3rd Level	0 (0)	3 (12.5)	11 (45.8)	6 (25.0)	2 (8.3)
		4th Level	5 (20.8)	11 (45.8)	1 (4.2)	2 (8.3)	5 (20.8)
		Worst Level	14 (58.3)	2 (8.3)	0 (0)	1 (4.2)	7 (29.2)
	Other	Best Level	0 (0)	0 (0)	0 (0)	0 (0)	2 (100.0)
		2nd Level	0 (0)	0 (0)	0 (0)	2 (100.0)	0 (0)
		3rd Level	0 (0)	0 (0)	2 (100.0)	0 (0)	0 (0)
		4th Level	0 (0)	2 (100.0)	0 (0)	0 (0)	0 (0)
		Worst Level	2 (100.0)	0 (0)	0 (0)	0 (0)	0 (0)

F.4 Correlation Between Ranking Results and Participant Profiles

Table F.11: Pearson's Chi Square Test results for participant groups

	Level 1	Level 2	Level 3	Level 4	Level 5
Age	$X^2(12, N=171)=27.16, p=0.007$	$X^2(12, N=171)=30.792, p=0.002$	$X^2(12, N=171)=32.882, p=0.001$	$X^2(12, N=171)=16.446, p=0.172$	$X^2(12, N=171)=24.678, p=0.016$
Education	$X^2(16, N=171)=42.207, p=0$	$X^2(16, N=171)=42.045, p=0$	$X^2(16, N=171)=38.734, p=0.001$	$X^2(16, N=171)=50.59, p=0$	$X^2(16, N=171)=50.114, p=0$
Gender	$X^2(4, N=171)=5.923, p=0.205$	$X^2(4, N=171)=6.441, p=0.169$	$X^2(4, N=171)=7.465, p=0.113$	$X^2(4, N=171)=1.907, p=0.753$	$X^2(4, N=171)=2.848, p=0.583$
Level	$X^2(8, N=171)=3.138, p=0.925$	$X^2(8, N=171)=9.346, p=0.314$	$X^2(8, N=171)=10.282, p=0.246$	$X^2(8, N=171)=6.602, p=0.58$	$X^2(8, N=171)=6.649, p=0.575$
Status	$X^2(12, N=171)=30.239, p=0.003$	$X^2(12, N=171)=26.214, p=0.01$	$X^2(12, N=171)=24.43, p=0.018$	$X^2(12, N=171)=29.14, p=0.004$	$X^2(12, N=171)=29.294, p=0.004$

APPENDIX G

DEMOGRAPHIC INFORMATION QUESTIONNAIRE

1. What is your gender?

- Female
- Male

2. What is your age?

- Under 18
- 18-24
- 25-34
- 35-54
- 55+

3. How often do you use the Web?

- Daily
- Weekly
- Monthly
- Less than once a month
- Never

4. How often do you use the Web on mobile environment?

- Daily
- Weekly
- Monthly
- Less than once a month
- Never

5. Which of the following best describes your current status involved with web design:

- I have worked in web design and development
- I have studied web design and development

- Web design and development is a hobby of mine (but not my area of employment/study)
- Other

6. How would you describe your level of expertise in web design and development?

- Professional
- Intermediate
- Novice/Beginner

7. Highest level of education you have completed:

- Grade/Primary School
- High/Secondary School
- Associate's Degree
- Bachelor's Degree
- Master's Degree
- Doctorate
- Other

APPENDIX H

USER EXPERIMENT QUESTIONNAIRE ITEMS

1. Overall, I am satisfied with the ease of completing this task.
2. Overall, I am satisfied with the amount of time it took to complete this task.
3. Overall, I am satisfied with the support information (on-line help, messages, documentation) when completing this task.
4. Overall, I am satisfied with how easy it is to use this system.
5. It is simple to use this system.
6. I could effectively complete the tasks and scenarios using this system.
7. I was able to complete the tasks and scenarios quickly using this system.
8. I was able to efficiently complete the tasks and scenarios using this system.
9. I felt comfortable using this system.
10. It was easy to learn to use this system.
11. I believe I could become productive quickly using this system.
12. The system gave error messages that clearly tell me how to fix problems.
13. Whenever I made a mistake using the system, I could recover easily and quickly.
14. The information (such as on-line help, on-screen messages and other documentation) provided with this system is clear.
15. It was easy to find the information I needed.
16. The information provided for the system was easy to understand.
17. The information was effective in helping me complete the tasks and scenarios.
18. The organization of information on the system screens was clear.
19. The interface of this system was pleasant.
20. I liked using the interface of this system.
21. This system has all the functions and capabilities I expect it to have.

22. Overall, I am satisfied with this system.

APPENDIX I

EVALUATION RESULTS OF WEB PAGE TRANSCODING

I.1 Item Based Test Results of Original and Transcoded Pages

Table I.1: Item based test results of original and transcoded pages

	Original			Transcoded		
	N	Mean	Std. Dev.	N	Mean	Std. Dev.
1. Overall, I am satisfied with the ease of completing this task.	25	5.840	1.179	25	6.240	1.091
2. Overall, I am satisfied with the amount of time it took to complete this task.	25	6.400	0.707	25	6.440	0.917
3. Overall, I am satisfied with the support information (on-line help, messages, documentation) when completing this task.	15	5.930	0.961	17	6.000	1.581
4. Overall, I am satisfied with how easy it is to use this system.	25	5.880	1.130	25	6.200	1.041
5. It is simple to use this system.	25	6.080	1.115	25	6.200	1.080
6. I could effectively complete the tasks and scenarios using this system.	25	6.160	1.214	25	6.280	0.891
7. I was able to complete the tasks and scenarios quickly using this system.	25	5.960	1.241	24	6.130	1.227
8. I was able to efficiently complete the tasks and scenarios using this system.	25	5.800	1.225	25	6.240	0.926
9. I felt comfortable using this system.	25	5.920	1.382	25	5.760	1.739
10. It was easy to learn to use this system.	25	6.000	1.291	25	6.440	1.121
11. I believe I could become productive quickly using this system.	24	5.380	1.663	24	5.830	1.606
12. The system gave error messages that clearly tell me how to fix problems.	5	4.600	1.949	7	2.290	1.704
13. Whenever I made a mistake using the system, I could recover easily and quickly.	13	5.080	1.935	8	5.380	1.685
14. The information (such as on-line help, on-screen messages and other documentation) provided with this system is clear.	13	6.150	1.214	11	6.000	1.000
15. It was easy to find the information I needed.	25	5.720	1.429	25	6.280	0.891
16. The information provided for the system was easy to understand.	23	6.220	0.902	25	6.520	0.653
17. The information was effective in helping me complete the tasks and scenarios.	24	6.290	0.955	25	6.080	1.382
18. The organization of information on the system screens was clear.	24	5.670	1.606	25	6.160	1.214
19. The interface of this system was pleasant.	25	5.760	1.451	25	6.000	1.414
20. I liked using the interface of this system.	25	5.280	1.696	25	5.640	1.729
21. This system has all the functions and capabilities I expect it to have.	24	5.380	1.583	24	5.580	1.613
22. Overall, I am satisfied with this system.	25	5.920	1.152	25	6.000	1.443

I.2 Transcoding Evaluation Results With Respect to User Profiles

Table I.2: Test results for demographic groups

			Original			Transcoded		
			N	Mean	Std. Dev.	N	Mean	Std. Dev.
Gender	Female	ASQ	14	6.095	0.862	11	6.424	0.593
		SYSUSE	14	5.662	1.326	11	6.409	0.528
		INFOQUAL	14	5.786	1.340	11	6.546	0.610
		INTERQUAL	14	5.804	1.418	11	6.000	1.215
	Male	ASQ	11	6.136	0.826	14	6.202	1.017
		SYSUSE	11	6.080	0.791	14	6.027	0.845
		INFOQUAL	11	6.023	0.637	14	6.107	1.208
		INTERQUAL	11	5.455	1.264	14	5.571	1.627
Age	18-24	ASQ	14	6.250	0.609	10	6.433	0.540
		SYSUSE	14	5.948	1.241	10	6.375	0.534
		INFOQUAL	14	6.071	1.124	10	6.450	0.654
		INTERQUAL	14	5.857	1.361	10	5.850	1.226
	25-34	ASQ	10	5.833	1.045	15	6.211	1.013
		SYSUSE	10	5.638	1.008	15	6.075	0.840
		INFOQUAL	10	5.600	1.049	15	6.200	1.185
		INTERQUAL	10	5.325	1.375	15	5.700	1.618
Web Usage in Mobile Devices	Daily	ASQ	20	6.182	1.004	21	6.302	0.920
		SYSUSE	20	5.899	0.975	21	6.244	0.765
		INFOQUAL	20	5.838	1.196	21	6.321	1.040
		INTERQUAL	20	5.546	1.364	21	5.833	1.419
	Weekly	ASQ	3	5.333	0.943	1	6.500	.
		SYSUSE	3	5.750	1.768	1	6.000	.
		INFOQUAL	3	5.917	0.144	1	6.250	.
		INTERQUAL	3	5.250	2.121	1	4.250	.
	Never	ASQ	2	6.500	0.707	2	6.333	0.471
		SYSUSE	2	6.438	0.619	2	6.250	0.530
		INFOQUAL	2	6.375	0.177	2	6.750	0.354
		INTERQUAL	2	6.500	0.707	2	6.875	0.177

Table I.3: Test results for demographic groups

			Original			Transcoded		
			N	Mean	Std. Dev.	N	Mean	Std. Dev.
Current Status Involved in Web Design and Development	Worked	ASQ	5	6.333	0.972	6	6.444	0.621
		SYSUSE	5	6.000	0.879	6	6.208	0.723
		INFOQUAL	5	6.150	0.877	6	6.542	0.781
		INTERQUAL	5	5.000	1.479	6	6.042	1.382
	Studied	ASQ	3	5.778	0.385	3	7.000	0.000
		SYSUSE	3	6.250	0.661	3	6.750	0.250
		INFOQUAL	3	6.000	0.250	3	7.000	0.000
		INTERQUAL	3	6.500	0.661	3	6.417	0.382
	Hobby	ASQ	6	6.028	0.687	4	6.458	0.712
		SYSUSE	6	5.982	0.683	4	6.344	0.449
		INFOQUAL	6	6.083	0.606	4	6.188	0.747
		INTERQUAL	6	5.875	1.034	4	6.063	0.826
	None	ASQ	11	6.152	0.970	12	6.000	1.003
		SYSUSE	11	5.591	1.498	12	6.000	0.861
		INFOQUAL	11	5.636	1.464	12	6.042	1.224
		INTERQUAL	11	5.591	1.526	12	5.354	1.766
Level of Expertise	Professional	ASQ	4	6.333	1.122	5	6.800	0.447
		SYSUSE	4	5.813	0.893	5	6.450	0.629
		INFOQUAL	4	5.938	0.851	5	6.900	0.224
		INTERQUAL	4	4.563	1.281	5	6.250	0.559
	Intermediate	ASQ	4	6.333	0.471	3	6.778	0.385
		SYSUSE	4	6.442	0.435	3	6.500	0.217
		INFOQUAL	4	6.688	0.375	3	6.500	0.500
		INTERQUAL	4	6.813	0.125	3	6.333	0.764
	Novice/Beginner	ASQ	17	6.010	0.847	17	6.069	0.913
		SYSUSE	17	5.713	1.262	17	6.066	0.807
		INFOQUAL	17	5.691	1.171	17	6.088	1.132
		INTERQUAL	17	5.632	1.317	17	5.515	1.671
Education	High/Secondary School	ASQ	10	6.233	0.649	3	6.278	0.255
		SYSUSE	10	6.189	0.844	3	5.833	0.520
		INFOQUAL	10	6.325	0.646	3	5.917	0.804
		INTERQUAL	10	6.200	0.919	3	4.750	1.561
	Bachelor's Degree	ASQ	11	6.167	0.916	19	6.386	0.724
		SYSUSE	11	5.727	1.379	19	6.342	0.579
		INFOQUAL	11	5.568	1.347	19	6.447	0.970
		INTERQUAL	11	5.341	1.652	19	6.171	0.954
	Master's Degree	ASQ	4	5.667	1.054	3	5.778	1.836
		SYSUSE	4	5.313	0.869	3	5.625	1.521
		INFOQUAL	4	5.688	0.944	3	5.750	1.392
		INTERQUAL	4	5.125	0.968	3	4.167	2.754

I.3 Correlation Between Transcoding Evaluation Results and Participant Profiles

Table I.4: Test results for participant groups (N: Normal, NN: Not Normal)

		ASQ		SYSUSE		INFOQUAL		INTERQUAL	
Gender	Female	N	t(23) = 1.079, p = 0.292	N	t(17.829) = 1.924, p = 0.071	NN	U = 50, p = 0.132	NN	U = 70.5, p = 0.718
	Male	NN	U = 68.5, p = 0.632	N	t(23) = 0.159, p = 0.875	NN	U = 58, p = 0.291	NN	U = 67.5, p = 0.601
Age	18-24	N	t(22) = 0.761, p = 0.455	NN	U = 63.5, p = 0.702	N	t(22) = 0.952, p = 0.351	NN	U = 67.5, p = 0.882
	25-34	NN	U = 55.5, p = 0.266	NN	U = 55, p = 0.266	NN	U = 40, p = 0.050	NN	U = 58, p = 0.342
Web Usage in Mobile Devices	Daily	NN	U = 105.5, p = 0.678	NN	U = 87, p = 0.257	NN	U = 148, p = 0.098	NN	U = 100, p = 0.536
	Weekly	NN	U = 0, p = 0.221	NN	U = 1, p = 1.000	NN	U = 0, p = 0.221	NN	U = 1, p = 1.000
	Never	NN	U = 1.5, p = 0.683	NN	U = 1, p = 0.439	NN	U = 0.5, p = 0.221	NN	U = 1.5, p = 0.683
Current Status Involved in Web Design & Development	Worked	NN	U = 15, p = 1.000	N	t(9) = 0.432, p = 0.676	NN	U = 9, p = 0.261	NN	U = 7, p = 0.140
	Studied	NN	U = 0, p = 0.034	N	t(4) = 1.225, p = 0.288	NN	U = 0, p = 0.037	N	t(4) = 0.189, p = 0.859
	Hobby	N	t(8) = 0.958, p = 0.366	N	t(8) = 0.925, p = 0.382	N	t(8) = 0.244, p = 0.814	N	t(8) = 0.302, p = 0.770
	None	NN	U = 58, p = 0.617	N	t(21) = 0.812, p = 0.426	NN	U = 55, p = 0.493	NN	U = 59, p = 0.663
Level of Expertise	Professional	NN	U = 7, p = 0.383	N	t(7) = 1.262, p = 0.248	NN	U = 1, p = 0.021	N	t(7) = 2.455, p = 0.071
	Intermediate	NN	U = 2.5, p = 0.190	NN	U = 6, p = 0.383	N	t(5) = 0.572, p = 0.592	NN	U = 3.5, p = 0.354
	Novice Beginner	NN	U = 135.5, p = 0.753	NN	U = 127.5, p = 0.557	NN	U = 106, p = 0.181	NN	U = 143, p = 0.958
Education	High/Secondary School	N	t(11) = 0.113, p = 0.912	N	t(11) = 0.680, p = 0.511	NN	U = 10.5, p = 0.439	NN	U = 6, p = 0.125
	Bachelor's Degree	NN	U = 88, p = 0.461	NN	U = 77, p = 0.235	NN	U = 47.5, p = 0.012	NN	U = 77, p = 0.231
	Master's Degree	N	t(5) = 0.102, p = 0.922	N	t(5) = 0.349, p = 0.742	N	t(5) = 0.072, p = 0.946	N	t(5) = 0.662, p = 0.537