Heuristic Role Detection of Visual Elements of Web Pages

M. Elgin Akpınar[1] and Yeliz Yeşilada[2]

[1] Middle East Technical University, Ankara, Turkey
[2] Middle East Technical University Northern Cyprus Campus, Kalkanlı, Güzelyurt, Mersin 10, Turkey
{elgin.akpinar,yyeliz}@metu.edu.tr

Abstract. Web pages are typically designed for visual interaction – they include many visual elements to guide the reader. However, when they are accessed in alternative forms such as in audio, these elements are not available and therefore they become inaccessible. This paper presents our ontology-based heuristic approach that automatically identifies visual elements of web pages and their roles. Our architecture has three major components: 1. automatic identification of visual elements of web pages, 2. automatic generation of heuristics as Jess rules from an ontology and 3. application of these heuristic rules to web pages for automatic annotation of visual elements and their roles. This paper first explains our architecture in detail and then presents our both technical and user evaluations of the proposed approach and architecture. Our technical evaluation shows that complexity is an important performance factor in role detection and our user evaluation shows that our proposed system has around 80% receptive accuracy, but the proposed knowledge base could be further improved for better accuracy.

Keywords: Web Accessibility, Knowledge Engineering, Ontology, Heuristics, Rule Engine, User Study.

1 Introduction

Web pages are typically designed for visual interaction, including many visual elements to guide the reader. As web technologies develop, they enable developers to create more unique and technically sophisticated web pages. However, far too little attention has been paid to accessibility issues. When pages are accessed in alternative forms such as in audio with assistive technologies, these visual elements are not available and therefore web pages become inaccessible.

In order to provide better accessibility, having deep understanding of the structure of web pages along with the role of visual elements is important. Once we identify the role of visual elements, these roles can be used to transcode the page by removing unnecessary elements or reorganise the page structure to improve the accessibility not only for disabled people but also for small screen device users. Furthermore, such understanding could also be used in improving the accuracy of information retrieval and data mining [1], better presenting pages for small screen devices [2, 3], or designing better intelligent user interfaces [4]. However, due to the flexible syntax of HTML and CSS,

F. Daniel, P. Dolog, and Q. Li (Eds.): ICWE 2013, LNCS 7977, pp. 123–131, 2013.

developers can create the same visual layout with different underlying coding, which means automating the role identification can be a challenging task [5]. There exist studies that propose a range of roles for visual elements, but the approaches tend to be simplistic and there is no work that provides a deep understanding of visual elements (Section 2). Some standards such as ARIA from W3C [6], and HTML5 do provide much stronger and better list of roles, but they still do not cover deep understanding of the visual elements of web pages.

This paper presents our ontology-based heuristic approach that automatically identifies visual elements of web pages and their roles (Section 3). It proposes an ontology that captures detailed knowledge about visual elements and presents an architecture with three main components. First of all, visual elements are automatically identified by using both the visual rendering and the underlying source code and a visual elements tree is generated. Then, heuristics are generated as rules from our ontology. Due to the flexible nature of HTML and CSS, we cannot fully and absolutely describe objects and their definite properties and do direct reasoning with the ontology over the visual elements. To address this problem, a probabilistic approach was constructed, in which, 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. Therefore, the proposed ontology describes the roles and their properties, which is then converted to rules. Finally, these heuristic-based rules are applied to visual elements for automatic annotation of their roles, generating a visual element tree with role assignments. Technical and user evaluations are also produced to validate the proposed approach and architecture (Section 4).

2 Related Work

There are different approaches for automatically identifying visual elements (some refers to them as segments or blocks) and discovering their roles [5] in different fields to serve different purpose.

Role identification has been proposed to better adapt web pages for small screen devices [2, 3, 7, 8]. Proposed roles tend to be simple lists which include roles such as lists content, related links, navigation and support, advertisement, etc. As opposed to these simple lists, Chen et al. [9] introduce a very comprehensive model called Function-Based Object model which hierarchically categorises objects as basic and composite objects. Even though the proposed model is very comprehensive, it does not take into account how an end-user would understand and use a web page.

Creating intelligent user interfaces also require good understanding of web page structures, meaning that it can more intelligently displayed on different devices and systems. Xiang and Shi [4] only proposes two kinds of visual elements: nontext blocks (buttons, images, inputs, etc) and text blocks. This simplistic approach does not provide a deep understanding of web page elements.

Role detection has also been explored for information retrieval and web data mining [1, 10–13]. Proposed roles include informative and redundant content blocks, heading, subtitle, paragraph, data, etc. Web accessibility is another field where role identification has been explored. Takagi et al. [14] proposes the following roles for

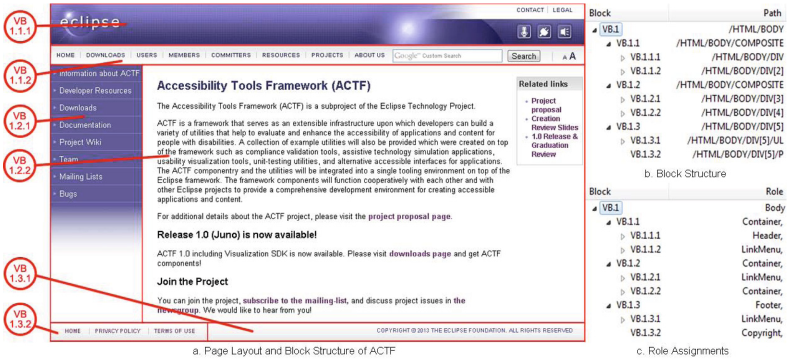

Fig. 1. The Home Page of ACTF

fragments: proper content, updated index, general index, norole, header, footer, advertisement, delete, layout table. Even though this looks like a comprehensive list, there still some roles that are missing. These roles also give the impression they were created to only guide the transcoding rather than semantically describing a web page.

In brief, discovering roles of visual elements of web pages is an important task in many fields. Even though, different research refers to different roles of visual elements, unfortunately, existing research typically focus on a small set of roles and there is no work that provides a deep understanding of visual elements.

3 Ontology Based Heuristic Role Detection

Our work aims to address the shortcomings and limitations of the existing work and provide a knowledge base that includes detailed information about visual elements of web pages and their properties. In our previous work, an ontology was created called ”WAfA (Web Authoring for Accessibility)” [15] to capture knowledge about visual elements which has been mainly used to annotate web pages manually [16] such that it can be used to re-engineer web pages for better accessibility for visually disabled people [17, 18]. However, there has been no work on automatically processing a web page to identify the roles specified in WAfA Ontology. The overall architecture of our proposed system to automate the process has the following components:

1. *Visual element identifier* takes a web page and uses its visual presentation and source code to automatically divide it into visual elements in a tree structure.
2. *Rule generator* component takes our knowledge base, implemented in an OWL ontology and generates heuristic rules for visual elements.
3. *Role detector* component takes rules and tree of visual elements, generated by our first component and returns a labeled tree of visual elements.

Our system has been implemented on the Accessibility Tools Framework (ACTF) of Eclipse Foundation[1].

[1] http://www.eclipse.org/actf/

3.1 Visual Element Identifier

In order to discover the roles of visual elements in a web page, we need to first divide a given web page into meaningful blocks to represent visual elements. There are many web page segmentation algorithms created for different purposes [5, 19–21]. In order to automatically segment a page, we have used and extended the Vision Based Page Segmentation Algorithm (VIPS) [5]. Compared to other approaches, VIPS uses both the underlying source code and visual cues which include font, size, color and tag attributes of the nodes. The segmentation task is processed in three main steps which are: visual block extraction, visual block separation and content structure construction [5]. We have implemented our extended VIPS algorithm on the ACTF platform. Once a web page is processed by our VIPS implementation, it produces a visual block tree (Figure 1 part b - "block structure"). The page is first divided into large blocks and then these blocks are further divided into a smaller blocks. In our implementation, the visual element identifier generates a tree visual elements and for each visual element an XPATH is also created.

3.2 Rule Generator

WAfA Ontology was created to capture shared understanding of visual elements of web pages [15]. Even though, WAfA is a very rich knowledge resource, our initial experiments with WAfA showed that, the characteristics of the concepts described in WAfA are too specific; therefore, automating the process based on such definitions may have resulted in false positive role assignments. Therefore, WAfA Ontology was selected as the base ontology and a new ontology, named as eMine Ontology [2], has been developed to be used for automated role detection.

Knowledge Representation. In order to systematically characterise the roles of visual elements and their properties, popular web pages, long-tail pages and also popular web pages from different genres listed in Alexa have been investigated and analysed. In these investigations, it was observed that the following properties are affecting how visual elements are used and presented: 1. underlying tag (HTML/HTML5 or ARIA [6]), 2. children and parent elements in the underlying DOM tree, size of the element, 3. border and background color of the element, 4. position of the element, 5. some attributes including onclick, for, onmouseover, etc., 6. CSS Styles (font-size, color, etc.) of the elements, 7. some specific keywords which appear in the textual content and in the id, class, src, background-image attributes of the element. In Figure 1, VB.1.1.1 represents a *Header* block. *Header* blocks are generally placed at the top of the page and contains a *Logo*, *Search Engine* or *Menu* block. In general use, HTML tag of a *Header* block is either header or div. Also some specific keywords such as 'header' or 'hdr' occur in their id or class attributes. It is possible to extend these characteristics.

The eMine ontology consists of two main classes: atom and chunk. All roles are defined as subclasses of either atom or chunk classes. In order to extend the coverage of the roles given originally in WAfA, eMine Ontology was also compared to ARIA Ontology [6], and similarities and differences between them were identified. After this analysis, some concepts were also introduced to eMine Ontology from ARIA.

[2] `http://emine.ncc.metu.edu.tr/ontology/emine.owl`

We mainly followed an iterative approach to develop this ontology. With the initial experiments, we have noticed that object properties are required to be classified. This classification takes in two parts. The first classification is between the object properties, so that, object properties which affect the decision more than other properties have higher factor values. The second classification is in the values of object properties. In this classification, three levels of object properties for each attribute are defined. For example, must_have_tag, has_tag and may_have_tag were required where must_have_tag has the highest factor value and may_have_tag has the lowest factor. In manual evaluation, it was observed that this kind of enhancement gave more accurate results.

The ontology was parsed and processed to construct heuristic rules to apply on visual elements. After retrieving all roles and their properties from the ontology, they were converted to Jess rules in appropriate syntax and saved in a local CLP file for later use in role detection [22].

3.3 Role Detector

For detecting the roles of visual elements, Jess which is a Java based rule engine and scripting environment, is used [22]. The reason of using Jess is that it provides a rule engine to assert the visual elements and fire a set of rules on asserted elements. After a web page is segmented to its visual elements, each element in the tree was accepted as an individual. Each individual visual element was converted to facts in Jess rule engine. The collection of the facts, is called working memory. Every fact has a template, which defines its name and the set of its slots. These slots keep the attributes of a fact, in this case, the visual element attributes corresponding to the object properties in the ontology.

In order to detect the heuristic role of a visual element, following process is applied: First of all, a rule engine object is created. By loading the CLP file into the working memory of the engine, block template for facts and likelihood scores for each role were constructed in the memory. Likelihood scores are defined as global variables, and initially set to 0. Then, a set of heuristic rules, which are called defrules, are defined. These rules are based on the object properties of roles in ontology. Following statement illustrates a rule definition for the role given for a Header block:

```
(defrule Header00 (block (has_child $? /.*logo.*/ $?))
                  => (bind ?*Header* (+ "2" ?*Header*)))
```

The first part after rule name denotes which object 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 contains a *Logo*. 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 2, which is the value of factor annotation for this rule.

Template definition, global variables and defrules are stored in the CLP file and they are all generated by using the eMine ontology. Facts which will be asserted to the rule engine, are based on the visual elements of segmented web pages and they are constructed in each program execution. This assertion and firing process is repeated for each block in the web page. After each iteration, visual element tree is updated to represent assigned role to visual elements. Figure 1 (part c - ”role assignments”) shows how the tree of visual element is annotated with the roles. For example, VB.1.1.1 is labelled as *Header*.

4 Evaluation

In order to validate our proposed approach, we have conducted both technical and user evaluations. Technical evaluation mainly investigates the performance and the user evaluation investigates how people perceive the role of visual elements, and how good is our system in identifying the perceived roles automatically.

User Evaluation. Our user evaluation[3] was conducted online to reach more with diverse backgrounds. The procedure followed in the survey included four main parts: 1. included an overview page with some information about the anonymity and the tasks to be completed. 2. collected some demographics information about participants, e.g., gender, experience in web design, education, age range, etc. 3. included a web page in different levels of segmentation, and participants were asked to rate and rank these levels. 4. Based on the best level they have chosen, they were asked to assign roles to the visual segments in that level. The participants were provided a list of roles in our knowledge base; however, if the participants could not find the proper role in the list, they also had the chance of entering the role in free form text. In overall, the survey application was designed to repeat the last two steps for randomly selected nine pages.

Table 1. Accuracy and Performance Results

Complexity Group	System-Expert Evaluation (%)	Receptive Evaluation (%)	Total Memory (KB)	Total Time (ms)	Avr. Memory per Block (KB)	Avr. Time per Block (ms)	Block Count
Low Comp.	79.82	73.68	8,369	6,576	244.29	102.29	65
Medium Comp.	88.28	79.77	7,013	23,799	36.44	102.12	237
High Comp.	88.47	85.53	9,165	54,837	34.28	101.95	569
Overall	86.83	80.82	8,176	29,157	100.20	102.11	298

The complete survey was designed to include nine randomly chosen web pages from a group of 30. In order to choose these 30 pages, we have investigated the complexity of top 100 web from Alexa by using the Visual Complexity Rankings and Accessibility Metrics (VICRAM) framework [23], which assigns a Visual Complexity Score (VCS) for a given page. For 100 pages, we calculated their VCS and grouped these pages into three: low complexity (VCS $<$ 3); medium complexity ($3 \leq VCS \leq 7$); and high complexity (VCS $>$ 7). We randomly selected 10 pages from each complexity level by grouping these pages based on their VCS. Moreover, ramdom page selection algorithm was designed to select at least one page from all complexity levels in three pages.

Technical Evaluation. With a technical evaluation, we have mainly investigated the technical feasibility of the proposed approach and implementation in the ACTF platform. We checked the performance characterised in terms of total memory usage, time elapsed for role detection of complete pages and total number of blocks calculated. 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 videocard and Windows 7 32 Bit operating system.

[3] `http://emine.ncc.metu.edu.tr/eval/survey/`

4.1 Results

In this section, we present the preliminary analysis of our results based on the data collected in two weeks after the survey was announced. In overall, of 220 participants, only 25 of them have completed at least three pages, provided that, they evaluated at least one page from each complexity level. Of our 25 participants, 10 were female and 15 were male. 7 participants were aged between 18-24, 8 of them were 25-34, 8 of them were 25-54 and 2 of them were aged over 55. 5 participants completed high/secondary school, 2 completed associate's degree, 6 completed bachelor's degree, 7 completed master's degree and 5 completed doctorate. 18 participants have worked in web design and development, 4 of them studied this subject and 3 of them are interested in web design as a hobby. 7 participants describe their level of expertise in web design and development as professional, 13 as intermediate and 5 as novice/beginner. All of the participants use internet daily.

In overall, 1,946 role assignments have been made and 1,458 were considered in our evaluation as valid assignments since their assigners satisfied the minimum requirement of labelling at least three pages. 232 roles assigned to low complexity pages, 580 to medium complexity pages and 646 to high complexity pages. Analyzing the assignments, we have noticed that there were disagreements between our participants. In order to eliminate this, we have applied majority rule to decide about the role assigned. When majority rule applied, %32.58 of the assignments had more than %50 of agreement.

Table 1 presents both performance results and accuracy results obtained from the preliminary analysis of the data collected from the participants. The success rate of each complexity group and overall result was calculated proportional to the number of visual segments 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. Strict string comparison between the roles assigned by the system and participants 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 to label visual elements. Therefore, we have manually analyzed the role assignments given by our participants and compared them with our system assignments. These results shown as "receptive evaluation" in Table 1. In overall, the accuracy rate is 80.82%.

Although, performance results presented in Table 1 is specific to our configuration of test machine, they still provide significant information about the overall performance. Total memory usage and time elapsed in role detection process of a whole page and a single block are given with the average block count for each complexity level.

4.2 Discussion

Our receptive evaluation shows that our proposed system has more than 80% accuracy rate; however, when we do strict string comparison this accuracy becomes around 30%. Moreover, majority rule application to the data collected from participants shows that, only %32.58 of the role assignments have the majority of participants' agreements.This could be explained by different reasons. First of all, people were not asked to complete tasks and they were just shown screenshots. Task specification could affect the role assignments. Furthermore, in our survey, we have explicitly asked participants to check the underlying source code of web pages and associated CSS files by providing links to the pages. Nevertheless, participants may not have analysed the DOM structure or interacted with segments; their overall assessment may have been formed only on the visual representation of the segments. One unanticipated finding from this evaluation was that, although we asked participants to choose only one role in our survey, some blocks have more than one role in page layout or they are combinations of different sub-blocks which have different roles. While some participants tend to assign multiple roles, many of them selected only one of them, omitting the remaining meaningful roles.

Average time elapsed for role detection of a single block has close values in each complexity level and total time is proportional to the number of blocks in the page. Total memory consumed for a page has close values for each complexity group, showing that, larger amount of the memory consumed for shared resources. Therefore, average memory consumed decreases while the number of blocks increases.

In summary, the results of this study suggest that, the roles of visual segments in web pages may differ with respect to the aim of usage and user point of view, since, the majority rule applies only on a small portion of the role assignments. Moreover, this study can be used to improve our knowledge base, by extending the role set according to responses of the participants.

5 Conclusion

This paper presented an ontology-based heuristic approach to automatically identify visual elements in a web page and their roles. This approach relies on a probabilistic model, in which, the role of the visual element is detected based on the number of its attributes which satisfy the requirements of a role. The proposed system consists of a visual element identifier, a knowledge base and a role detector module.

The proposed approach was evaluated with technical performance and user evaluations. According to results in performance evaluation, response time is related to the complexity of the page, while memory consumption is independent of the complexity if shared resources are used. In order to measure the accuracy of the system, an online survey based user evaluation was performed which shows that our proposed system has around 80% receptive accuracy; however, the proposed knowledge base could be further improved for better accuracy.

In conclusion, the research presented in this paper contributes an effective method for detecting roles of visual elements in web pages automatically by using heuristics. The findings of this research can be used in different fields including information retrieval, web accessibility, intelligent web user interfaces, web page transcoding or data mining. Proposed approach also provides a modifiable knowledge base to adapt changing web design trends and task specific applications.

References

1. Kovacevic, M., Diligenti, M., Gori, M., Milutinovic, V.: Recognition of common areas in a web page using visual information: a possible application in a page classification. In: ICDM 2002, pp. 250–257. IEEE Computer Society, Washington, DC (2002)

2. Yin, X., Lee, W.S.: Understanding the function of web elements for mobile content delivery using random walk models. In: WWW 2005, pp. 1150–1151. ACM (2005)
3. Chen, Y., Xie, X., Ma, W.Y., Zhang, H.J.: Adapting web pages for small-screen devices. IEEE Internet Computing 9(1), 50–56 (2005)
4. Xiang, P., Shi, Y.: Recovering semantic relations from web pages based on visual cues. In: IUI 2006, pp. 342–344. ACM (2006)
5. Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: Vips: a vision based page segmentation algorithm, MSR-TR-2003-79, Microsoft Research (2003)
6. Craig, J., Cooper, M.: Accessible rich internet applications (WAI-ARIA) 1.0 (2010), `http://www.w3.org/TR/2010/WD-wai-aria-20100916/complete` (retrieved on January 15, 2013)
7. Ahmadi, H., Kong, J.: Efficient web browsing on small screens. In: AVI 2008, pp. 23–30. ACM (2008)
8. Xiao, Y., Tao, Y., Li, W.: A dynamic web page adaptation for mobile device based on web2.0. In: ASEA 2008, pp. 119–122. IEEE Computer Society, USA (2008)
9. Chen, J., Zhou, B., Shi, J., Zhang, H., Fengwu, Q.: Function-based object model towards website adaptation. In: WWW 2001, pp. 587–596. ACM (2001)
10. Lin, S.H., Ho, J.M.: Discovering informative content blocks from web documents. In: SIGKDD 2002, pp. 588–593. ACM (2002)
11. Burget, R., Rudolfova, I.: Web page element classification based on visual features. In: ACIIDS 2009, pp. 67–72 (April 2009)
12. Liu, B., Chin, C.W., Ng, H.T.: Mining topic-specific concepts and definitions on the web. In: WWW 2003, pp. 251–260. ACM (2003)
13. Yi, L., Liu, B., Li, X.: Eliminating noisy information in web pages for data mining. In: SIGKDD 2003, pp. 296–305. ACM (2003)
14. Takagi, H., Asakawa, C., Fukuda, K., Maeda, J.: Site-wide annotation: reconstructing existing pages to be accessible. In: SIGACCESS 2002, pp. 81–88. ACM (2002)
15. Harper, S., Yesilada, Y.: Web authoring for accessibility (WAfA). JWS 5(3), 175–179 (2007)
16. Yesilada, Y., Harper, S., Goble, C., Stevens, R.: Screen readers cannot see. In: Koch, N., Fraternali, P., Wirsing, M. (eds.) ICWE 2004. LNCS, vol. 3140, pp. 445–458. Springer, Heidelberg (2004)
17. Plessers, P., Casteleyn, S., Yesilada, Y., Troyer, O.D., Stevens, R., Harper, S., Goble, C.: Accessibility: A web engineering approach. In: WWW 2005, Chiba, Japan, pp. 353–362 (2005)
18. Yesilada, Y., Stevens, R., Harper, S., Goble, C.: Evaluating DANTE: Semantic transcoding for visually disabled users. ACM TOCHI 14(3) (2007)
19. Alcic, S., Conrad, S.: Page segmentation by web content clustering. In: WIMS 2011, pp. 24:1–24:9. ACM (2011)
20. Kohlschütter, C., Nejdl, W.: A densitometric approach to web page segmentation. In: CIKM 2008, pp. 1173–1182. ACM (2008)
21. Yu, S., Cai, D., Wen, J.R., Ma, W.Y.: Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In: WWW 2003, pp. 11–18. ACM (2003)
22. Friedman-Hill, E.: Jess the rule engine for the java platform (2008), `http://herzberg.ca.sandia.gov/` (retrieved on November 27, 2012)
23. Michailidou, E.: ViCRAM: Visual Complexity Rankings and Accessibility Metrics. PhD thesis (2010)