Hindawi Publishing Corporation
Advances in Human-Computer Interaction
Volume 2014, Article ID 479286, 13 pages
http://dx.doi.org/10.1155/2014/479286

Research Article

An Intelligent Framework for Website Usability

Alexiei Dingli and Sarah Cassar

Department of Intelligent Computer Systems, Faculty of ICT, University of Malta, Malta

Correspondence should be addressed to Alexiei Dingli; alexiei.dingli@um.edu.mt

Received 8 June 2013; Revised 30 December 2013; Accepted 13 January 2014; Published 14 April 2014

Academic Editor: Kerstin S. Eklundh

With the major advances of the Internet throughout the past couple of years, websites have come to play a central role in the modern marketing business program. However, simply owning a website is not enough for a business to prosper on the Web. Indeed, it is the level of usability of a website that determines if a user stays or abandons it for another competing one. It is therefore crucial to understand the importance of usability on the web, and consequently the need for its evaluation. Nonetheless, there exist a number of obstacles preventing software organizations from successfully applying sound website usability evaluation strategies in practice. From this point of view automation of the latter is extremely beneficial, which not only assists designers in creating more usable websites, but also enhances the Internet users' experience on the Web and increases their level of satisfaction. As a means of addressing this problem, an Intelligent Usability Evaluation (IUE) tool is proposed that automates the usability evaluation process by employing a Heuristic Evaluation technique in an intelligent manner through the adoption of several research-based AI methods. Experimental results show there exists a high correlation between the tool and human annotators when identifying the considered usability violations.

1. Introduction

Since its early years of existence, the Internet has made massive advances to transform into a form we know of today. With this development, the business marketing program has undergone parallel changes, making the Internet the most prominent channel for such a purpose [1–5]. In turn, this marketing shift has imposed a need on professionals to maintain a website for their business, which plays a very central role in this respect. However, owning a website is not enough for a business to be successful on the Web. In fact, in the not-so-distant past, a significant number of companies failed at transporting their business to the online environment simply because they disregarded the usability of their websites [6].

1.1. Usability and Accessibility. A general definition of usability is given by the International Standards Organization's ISO9241 standard, which states that "Usability is the" extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use [7].

On the other hand, the World Wide Web Consortium [8] defines accessibility on the Web as an attribute by which "people with disabilities can perceive, understand, navigate, and interact with the web, and they can contribute to the web." It is clear that, by definition, accessibility is aimed at abolishing any limitations encountered by people with any type of disability, including visual, auditory, physical, speech, cognitive, and neurological disabilities so as to make the content on the Web accessible to anyone.

Despite the technical differences in meaning between both terms, there is some confusion when it comes to usability and accessibility. In reality, accessibility is a subset of usability. As a matter of fact, a website is not usable unless it is accessible [9]. Branjik [10] explains this relationship by articulating that "while usability implies accessibility (at least when an unconstrained user population is considered), the contrary is not necessarily true."

1.2. The Importance of Usability on the Web. Regarding the Web, Nielsen [11] classifies usability as a quality attribute and defines it through five quality components: (1) *learnability*—the ease with which first-time users can manage to exercise all basic functionalities of the design; (2) *efficiency*—the speed with which users can carry out their tasks once they are accustomed to the design; (3) *memorability*—the ease with which users can resume their former skills of site usage; (4) *errors*—the frequency, severity, and ease of recovery from user-made errors; and (5) *satisfaction*—the enjoyment of using the design. He stresses the importance of website usability by claiming that it is "a necessary condition for survival," since its absence is very likely to frustrate and confuse users leading them to abandon that website for another competing one [12, 13].

On similar grounds, Fisher et al. [14] report that the quality of navigation and ease of site usage influence the amount of site content that is actually read, the users' emotional response to the website, the users' frustration, and their intention to revisit that website. Moreover, when a website is regarded as highly usable, users unknowingly increase their trust [15–17] and consequently their loyalty [17] towards that company. It has also been observed that usability is a major influence on user satisfaction [2, 17, 18] and encourages future revisits to the website [18].

1.3. Website Usability Guidelines. Several researchers have made an attempt at identifying which elements contribute towards good website design and usability. This led to the emergence of quite a few usability guidelines, or heuristics, that have been formulated both for generic user interfaces and for webpage design. Examples include those developed by Smith and Mosier [19], Norman [20], Nielsen [21], Comber [22], Sano [23], Borges et al. [24], Spool et al. [25], Fleming [26], Rosenfeld and Morville [27], Shneiderman [28], Nielsen [29], Dix et al. [30], and Nielsen and Loranger [12].

Despite the numerous website usability guidelines that have been developed throughout the years, there is currently no guideline set that has been established as a standard guiding framework [30]. As a means of addressing this problem, Mifsud [31] proposed a set of 242 research-based website usability guidelines compiled on the basis of the results from other usability studies carried out by researchers and experts in the fields, including [9, 12, 32–34].

1.4. Usability Evaluation (UE) and the Need for Automation. Being a software quality attribute, the usability of a design is not achieved through wishful thinking. Thus, a thorough evaluation is necessary to ensure an acceptable level of usability is attained [35], which has been shown to increase sales [36], competitiveness [37], user throughput [38], and user satisfaction [37], whilst decreasing training budgets [39] and needs for user support [40]. There is a general consensus regarding the common activities involved in the process of UE [41–43], which are, namely, the following:

(i) capture—the collection of usability data, such as "task completion time, errors, guideline violations, and subjective ratings";

(ii) analysis—the interpretation of the previously collected usability data with the aim of identifying usability problems in the design;

(iii) critique—the provision of suggestions in an attempt to alleviate the previously identified problems and improve the usability of the design.

Several methods exist to carry out these tasks, which can be very broadly categorized into two groups [30, 44, 45], namely, evaluation through user participation, also referred to as empirical testing [46, 47], and evaluation through expert analysis, also referred to as inspection methods [46–48].

Empirical testing is a form of usability testing that requires the participation of real users [30, 44, 47, 48]. Participants are asked to interact with the system so that their behaviour and the system's response are observed. They are additionally requested to offer suggestions for improvement of the design and its usability. This UE method typically takes place during advanced stages of development, where there exists at least a working prototype of the system so that user interaction is possible [30, 47, 49]. Some empirical testing methods include Protocol Analysis [50], Think Aloud [30, 51, 52], Post-Task Walkthroughs [30], Interviews, and Questionnaires [30, 51].

On the other hand, *inspection methods* aim at surfacing usability problems in a design without the involvement of users [30, 44, 47, 48]. Thus, they can be exercised during the early stages of development [30, 46, 49]. Through inspection, an interface designer or usability expert assesses a design for conformance to a set of predefined usability guidelines [30, 44]. The most common inspection technique is Heuristic Evaluation defined as "the most informal method and involves having usability specialists who judge whether each dialogue element follows established usability principles" [46]. An attractive characteristic of this technique is its capability of detecting the majority of usability problems encountered in a design [45, 53–55].

Despite the fact that software organisations in general have started to recognize the significance of usability [56], there exist several obstacles preventing them from successfully applying sound evaluation strategies in practice:

(i) UE methods are expensive in terms of time [37, 41, 57, 58] and human resources [37, 41–43, 58, 59], and thus it is not always possible to analyse every aspect of a design and/or compare several design alternatives [41];

(ii) it is difficult to find usability experts [43, 54, 59] and users belonging to the target group of the system [37, 58, 59];

(iii) usability findings suffer from subjectivity [54, 60, 61], which leads to results being nonsystematic or nonpredictable [41];

(iv) the mind-set of developers and their main focus and interest is the efficiency of code and system functionality. This completely diverges from users' judgements and concerns, thus obstructing the evaluation for usability [37, 58];

(v) software organizations do not truly know the meaning of usability and are unaware of the vast methods for its evaluation [37, 58].

1.5. Automated Usability Analysis Tools. Currently, there exist a number of tools which perform website usability analysis. Matera et al. [62] explain that such tools analyse the presentation layer to discover problems related to content presentation and navigation commands. It is important to note that some of these tools only focus their analysis on the accessibility of a website, as opposed to its usability in general. For this reason, we shall make the following distinction between these tools as (a) accessibility analysis tools and (b) usability analysis tools.

1.5.1. Accessibility Analysis Tools. MAGENTA [63] (Multi-Analysis of Guidelines by an Enhanced Tool for Accessibility) is a web-based accessibility tool developed by the Human Interface in Information Systems (HIIS) within the Human Computer Interaction Group. This tool not only references the Web Content Accessibility Guidelines-WCAG 1.0 [64] guidelines, but also evaluates the accessibility of websites according to their conformance to guidelines for the visually impaired and guidelines included in the Stanca Act [65]—the Italian accessibility law. MAGENTA identifies accessibility problems and where possible provides corrections of the identified accessibility violations [66].

Similarly, OCAWA [67] is a web-based automated accessibility evaluation tool developed by Urbilog and France Telecom. It refers to WCAG 1.0 [64] and France's accessibility law, the RGAA-Référential Général d'Accessibilité pour les Administrations [68]. Users can submit the URL of the website or upload an HTML file and the tool displays an accessibility audit report with links to the discovered violations [42, 67].

Likewise, WebA [55] (WebAnalysis), developed by the Aragonese Laboratory of Usability, references the WCAG 1.0 guidelines and partially performs some usability evaluation by assessing a website's adherence to the ISO 9241-11 to 17 Norms [7]. It also provides a test to measure user satisfaction and an application for card sorting, called aCaSo that enables users to participate in card sorting sessions.

1.5.2. Usability Analysis Tools. There are two types of tools that can perform automated usability evaluation (i) those that try to predict the usage of websites; and (ii) those that make use of conformance to standards [69].

(i) Tools Predicting the Usage of Websites. The Cognitive Walkthrough for the Web (CWW) is a modification on the theory-based usability inspection method of Cognitive Walkthrough (CW) [70–72] that makes use of Latent Semantic Indexing techniques to estimate semantic similarity for calculating the information scent of each link [69]. Through the use of CWW, the design team is provided with theoretical suggestions of what the use's next heading/link selection might be [73]. However, the use of CWW is a cumbersome process as it is not able to automatically analyse all the pages of a website and so requires manual intervention for completion of the analysis [69].

The InfoScent Bloodhound Simulator [69] is an academic prototype that performs automated usability evaluation of websites through the use of information scent. This technique makes use of the "Law of Surfing" and "Information Foraging Theory" to predict a user's surfing pattern.

WebCriteria SiteProfile makes use of usability metrics retrieved by browsing agents which navigated across websites [41, 62, 74]. These agents make use of a model based on GOMS (Goals, Operators, Methods, and Selection rules) to retrieve data which they integrated into usability metrics with the aim of assessing page loading time and the ease with which content is located [75, 76].

In the literature, the method by which usability metrics are retrieved by the agents is criticized [69] due to their performance when traversing a website based on its hyperlink structure without considering content analysis and aspects such as satisfaction of information needs, perception of navigation choices, and decision making ability—all of which are critical factors that affect the way real users navigate [69, 75]. In addition, Brajnik [10] states that WebCriteria SiteProfile only addresses a small set of usability attributes such as download time, alt text for images, and HTML validation, whilst it completely ignored other aspects such as consistency and information organisation.

Web TANGO (Web Tool for Assessing Navigation & Organisation) is a prototype that utilizes the Monte Carlo simulation and information retrieval algorithms to predict a user's information seeking behaviour and navigation paths through a website [77, 78]. In its evaluation it considers up to 157 highly accurate, quantitative metrics [79] such as colour and placement of text that are derived from the analysis of over 5,300 webpages [80]. It then compares the findings against empirically validated counterparts from successful sites and sites that were nominated for the Webby awards and received high ratings from the judges [69, 81, 82] so as to calculate their Webby Score [76].

A limitation of Web TANGO is its focus on the evaluation of individual webpage design issues rather than navigation and the website's information architecture [69]. In fact, it has been articulated that this tool is just a rudimentary system that is not robust enough to be adopted for a wider use [78]. Furthermore, Montero et al. [79] state that, out of the 157 metrics considered, only 6 actually assess the usability of a website.

(ii) Tools Making Use of Conformance to Standards. USEFul (Usability Evaluation Framework) is a web-based, nonpublically available prototype proposed by Mifsud [31] that evaluates the usability of a website by employing the Heuristic Evaluation technique which references the set of research-based usability guidelines compiled by the same author. It allows scalability for future enhancements as it separates the research-based usability guidelines from the evaluation logic that references them. In doing so, it allows a user to add, modify, or delete guidelines without altering the code.

Although USEFul is capable of referencing a substantial number of the 242 research-based guidelines, it only considers those that are somewhat straightforward to implement. During evaluation, it does not take into account the CSS of a website and any of its javascript files.

2. Aims and Objectives

As a means of assisting website designers in developing usable websites, the aim of this paper is to present a tool which automates the process of website usability evaluation in order to eliminate the current obstacles preventing this process from being exercised in practice. Although a number of tools have already been proposed in this field, the ideal one should

(i) *be located online and accessible as a web application* to mainstream the process of usability evaluation by targeting a larger audience and reduce the costs associated with installations and logistics of local systems;

(ii) *fully automate the capture, analysis, and critique UE activities* to be independent of human intervention;

(iii) *employ the Heuristic Evaluation technique* for its ability of surfacing the majority of usability problems encountered in a design through the inspection of a set of research-based website usability guidelines compiled in [31];

(iv) *collect and present evaluation results* in the form of user-friendly reports to aid users gain insight into the usability of their websites.

The only tool reported in the literature which meets these requirements is USEFul. However, this tool fails at handling guidelines which are somewhat sophisticated in nature and require advanced techniques to be automatically inspected. Thus, the focus of this paper is now reduced to tackle this problem, that is, to find methods which allow the automatic inspection of such guidelines and in doing so enhancing the capability of USEFul.

Since there is no access to this tool, a new prototype is developed as an online Java web application, termed the *Intelligent Usability Evaluation* (IUE) tool, which satisfies the imposed requirements and considers the following guidelines that have been chosen on the basis of their sophisticated nature, and due to the fact that they are included in the research-based set of guidelines developed by Mifsud, but are not referenced by USEFul:

(i) the headings that are used should be unique from one another and conceptually related to the content (HEADINGS_G guideline);

(ii) the use of graphical text should be kept to a minimum (GRAPHICAL_G guideline);

(iii) the homepage should look visually different from other webpages, whilst maintaining a consistent look-and-feel throughout the website (HOMEPAGE_G guideline).

The IUE tool is a proof-of-concept, or prototype, that primarily targets websites accessed through desktop computers. But the same methods and techniques can be easily used to evaluate mobile websites; however, this is not covered in this paper.

3. Design and Implementation

A fundamental aspect of the proposed IUE tool is to minimize the amount of human intervention necessary to evaluate a website for usability. Accordingly, the only user input required by the IUE tool is merely the homepage URL of the website to be evaluated and an e-mail address to send the generated reports. Having entered a homepage URL, the user issues an AJAX request to have a list of interior webpages automatically extracted from the homepage to be evaluated for usability. By default, the usability evaluation of a website is carried out by inspecting all the implemented guidelines. However, the user is free to restrict this inspection scheme by explicitly selecting which of these guidelines the tool should reference.

On completion of the evaluation process, results are collected and presented in the form two PDF reports that differ in the grouping of these results: *by guideline* or *by webpage*. By looking at these two reports individually, the user gains insight into which guideline is mostly breached and which webpage is violating the most guidelines, respectively. In order to make these reports as user-friendly as possible, rating images are used to facilitate the interpretation of the evaluation results. Moreover, suggestions of improvement are given in the form of *usability tips* that the user should relate to in case of guideline violations. Due to a possibly lengthy evaluation process, the user is not kept waiting online for the generation of these reports. Rather, they are sent as download links to the e-mail address provided so that the user can access them at any desired time once the evaluation is complete.

So as to achieve these functionalities, the system is decomposed as illustrated in Figure 1. User interaction with the tool occurs at the *presentation layer*, whose responsibility is to format and display data to the user. This layer requires services offered by the *business logic layer*. In particular, validation of the evaluation-form input depends on the *form-filling services* and fulfilling a user request to view the generated reports entailing the *reports view services*.

Most importantly, however, is the *usability evaluation service* that is accountable for the automatic usability evaluation of a website, which is invoked when the user issues a valid evaluation request. As stated in Section 2, this process should be carried out through an automatic Heuristic Evaluation, and thus it involves the execution of processes responsible for inspecting the relevant guidelines. These inspection processes are the most fundamental procedures of the tool and are the main focus of the next section.

3.1. Automatic Guideline Inspection.

Methods employed for the automatic guideline inspection processes are based on research findings as a result of fulfilling the first objective of this paper.

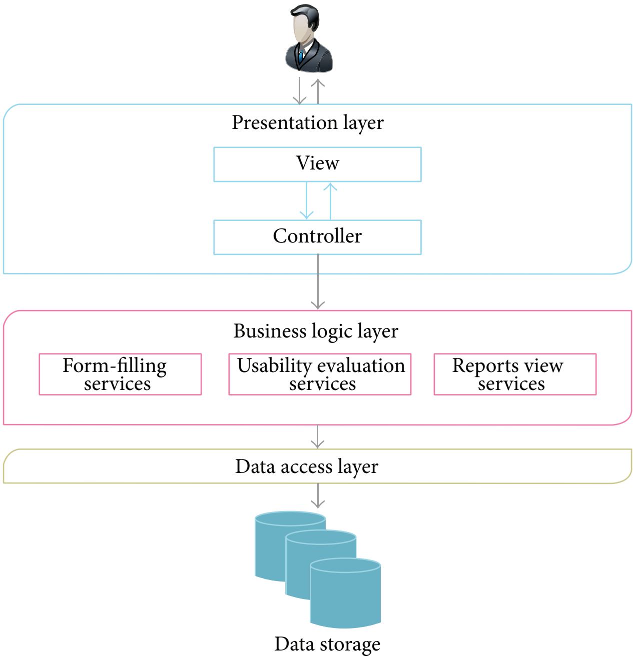


Figure 1: A high-level architectural design of the IUE tool.

3.1.1. HEADINGS_G Guideline Inspection. Automatic inspection of the HEADINGS_G guideline demands a solution that is able to (a) *extract the heading structure* of a webpage; (b) measure the *uniqueness* of every pair of extracted webpage headings; and (c) measure the *semantic relatedness* of all extracted heading text against their content text. As a result, the following methodology is adopted.

(a) A rule-based approach following the method proposed in [83] is undertaken to automatically extract the headings of a webpage in the form of a list with heading levels assigned through a numbering scheme. The general setting of this procedure consists of three main steps.

(i) *Identifying the general features of a webpage*. This necessitates a DOM tree traversal to extract text-formatting features from every text node. The general features of a webpage are consequently determined by counting the number of words in each text node and choosing those styles that span the most words.

(ii) *Identifying candidate heading nodes*. After cleaning a webpage from its boilerplate and unwanted content, a list of candidate headings can be formulated from those DOM tree nodes whose features are more significant than the general features of the webpage.

(iii) *Refining the list of candidate headings*. The list of candidate headings is put through the following refinement processes:

(1) remove candidate headings whose text is not sufficiently limited in length, ends with a punctuation mark, or does not start with a number or capitalized letter;

(2) remove candidate headings with equivalent styling features for their heading text and their content text;

(3) remove candidate headings which are not followed by any content;

(4) remove candidate headings whose heading text has weaker styling features than the content text.

(b) For every pair of extracted headings, the degree of *uniqueness* is measured by means of the N-Gram similarity measure for its ability to detect partial matches in addition to exact word matches that are also significant for this task. The similarity score obtained by the N-Gram measure, $\text{SIM}_{\text{N-Gram}}$, is transformed into a distance score to denote uniqueness by $1 - \text{SIM}_{\text{N-Gram}}$.

(c) For every extracted heading, the semantic relatedness of the heading text against the content text is measured via Explicit Semantic Analysis [84] which exploits Wikipedia as its background knowledge and is the current state-of-the-art semantic relatedness measure.

3.1.2. GRAPHICAL_G Guideline Inspection. The problem of automatically detecting graphical text in images calls for an image classifier specifically trained to distinguish between *graphical text images* and *nongraphical text images*. Note that a likely solution to this problem would involve the adoption of OCR software, with which an image is very easily classified as a *nongraphical text image* when no textual content is detected. However, the identification of text in an image could only serve as a possible indicator of graphical text since, very often, webpage images do contain appropriate snippets of text (e.g., Figure 2). This suggests that the use of OCR is not suitable for detecting graphical text in images.

For this purpose, the nonparametric *Naïve-Bayes Nearest Neighbour* (NBNN) [85] image classifier is employed for its simplicity, efficiency, avoidance of a learning/training phase, and its comparable performance to the top-leading learning-based approaches. The job of this classifier is to assign an image with one of three classes—*image*, *text*, or *mixture*—according to its contents. The *mixture* class is intended so that the classifier can distinguish between images that inherently contain a percentage of text and images which abuse this privilege by incorporating large chunks of text styled indifferent from the definite text of a webpage in addition to some image content.

Forty-five images collected from the EBay, Wikipedia, and Amazon webpages were manually classified as *image, text,* or *mixture* images to construct a sample dataset consisting of fifteen images per class on which the classifier can base its classification decisions. For every class, the sample images are preprocessed to extract their edge image by means of the Canny edge detection algorithm. Local SIFT descriptors are then computed for each detected region of interest in

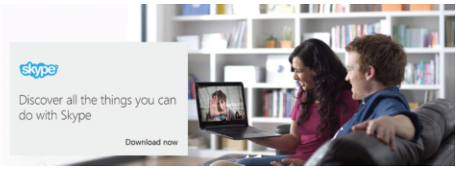


Figure 2: A webpage image containing an appropriate snippet of text (source: Microsoft.com).

the edge image and stored in an arff file. During classification, these files are loaded into kd-trees to improve the run-time and computational complexity of the algorithm. An observed image is preprocessed congruent to the sample images to get hold of its SIFT descriptors. The nearest neighbour of each descriptor to the three classes, respectively, is calculated and used for the computation of the *image-to-class distance* to return the class $\hat{C}$ which minimizes this value.

3.1.3. HOMEPAGE_G Guideline Inspection. Automatic inspection of the HOMEPAGE_G guideline demands a solution that is able to (a) measure the degree of similarity in terms of the perceived structural layout of the webpages and (b) check for a consistent look-and-feel throughout the pages of a website. To meet these requirements, the following methodology is adopted.

(a) With every webpage there are associated two representations: a *textual* representation as encoded in the source file and a *visual* representation as rendered by web browsers. Either one of these representations can be used as the basis upon which visual similarities can be compared. It is instinctively understandable as to why researchers opt for making use of the rendered web page image. The reason is that such representations correspond directly to what users view on a web browser, and thus comparison methods built upon them measure the actual visual similarity as perceived by users [86]. However, web browsers display requested pages by parsing HTML tags and CSS scripts and executing JavaScript code [87]. It is therefore evident that the rendered web page image relies in its entirety on the textual source file. As a result, some researchers decide on analysing the underlying source code in order to compare the visual similarities of web pages [88]. Despite being a common approach among researchers, visual similarity computation based on the source code is not an optimal solution.

(1) Using visual information exhibited by the rendered web page is more generic than that derived from analysing the underling HTML code [89]. The fundamental reason is that different HTML tagging sequences may possibly have the same visual effect [86, 90–92]. Furthermore, the growing tendency toward dynamic web technologies is affecting the amount of information contained in HTML pages with regard to their content [93]. In point of fact, HTML pages are nowadays lacking in this kind of information, at times containing almost none at all, as in the case of Flash-based websites.

(2) The majority of web pages are built "for human eyes" [90]. Consequently, it makes more sense to use the actual web page image in contrast to the source code. The benefit of doing so is that subsequent similarity comparisons actually measure the perceived similarity by the users rather than similarities of the code [86].

For this reason, an approach adapted from the method introduced in [88] is employed to measure structural layout similarities between two webpages on the basis of their rendered images. The general setting of this technique consists of four main steps:

(3) Segment the rendered webpage images into visually consistent regions. This process is carried out through the segmentation algorithm proposed in [94] whose basic foundation is rooted in two features unique to webpages which relate to the fact that every HTML element is displayed in its own rectangular area and the visible elements of a webpage are separated by background space. This means that every visually consistent region of a webpage corresponds to a rectangular area on the rendered webpage image. Based on these characteristics, the algorithm is composed of two major steps: (1) preprocess the rendered webpage image by means of the Canny edge detection algorithm to reduce its complexity and retain only the relevant information and (2) iteratively divide and shrink the edge image into indivisible subimages (regions).

(4) Transform the segmented webpage images into visual feature models. Visual feature models represent complete Attributed Relational Graphs (ARGs) with attributes assigned to both the nodes and the edges of the graph as defined in [88].

(5) Compute the Graph Edit Distance (GED) of the resulting visual feature models through a bipartite graph matching technique introduced in [95] by means of the Hungarian algorithm. The Hungarian algorithm is originally intended to efficiently solve the assignment problem. However, the graph matching problem can be translated into an assignment problem as follows: "How can one assign the nodes of graph G1 to the nodes of graph G2, such that the overall edit costs are minimal?" which makes it possible to adopt the Hungarian algorithm for the computation of the GED.

(6) Bound the GED score to a value between 0 and 1 by a normalization ration defined as $(||V1| - |V2||) + (||E1| - |E2||) + |V1| + |E1|$ where G1 and G2 are two ARGs such that $|V1| = n \leq m = |V2|$. Since the GED is a distance measure, it can be easily converted into a similarity measure as 1–GED.

(b) Determining the consistency of a website's look-and-feel is inspired by the method in [96, 97] and accordingly consists of the following procedures.

(1) *Compare the overall visual appearance* of every pair of webpages by computing their colour histograms from the rendered images and taking the intersection to obtain a similarity score between 0 (perfect mismatch) and 1 (perfect match).

(2) Find similarities between two font sizes f_1 and f_2 (expressed in pixels) as $1 - |f_1 - f_2|/\max(f_1, f_2)$.

(3) Find similarities between font-color and background-color RGB colours (r_1, b_1, g_1) and (r_2, b_2, g_2) as $1 - (1/(3 \times 255)(|r_1 - r_2| + |b_1 - b_2| + |g_1 - g_2|))$.

(4) Find similarities between font-family styles finally by checking if one font-family declaration is a substring of the other.

An $n \times m$ similarity matrix is built for the pairwise font-size, font-colour, and font-family similarities and used by the Hungarian algorithm to determine the most-matching pairs of the matrix, sum up the individual similarity scores, and normalize by the minimum of n and m for every matrix to obtain a single score bounded between 0 (perfect mismatch) and 1 (perfect match).

4. Results and Evaluation

A total of six experiments were conducted to assess the effectiveness of the IUE tool as an automatic website usability evaluator from the point of view of its *ability to detect guideline violations*. In the absence of website usability experts, the aim of the evaluation experiments is to determine *how well the tool correlates with human judgement* because usability is fundamentally concerned with the end users of a system and thus should strive to meet their verdicts.

For this reason, six *golden standard datasets of correct answers* were manually annotated by human judges, one for every experiment. This is necessary because each experiment strives to assess a different aspect of the guideline inspection processes. If the tool is shown to adequately model human judgement during inspection by comparing its results against the golden datasets, then it can be concluded that it is well suited for the task of website usability evaluation.

The conducted experiments are reported in the following sections grouped under the guideline they attempt to evaluate:

4.1. GRAPHICAL_G Inspection Evaluation Experiments

4.1.1. Experiment 1

Aim. to assess the classification performance of the employed NBNN image classifier for the task of detecting graphical text in webpage images on the dataset using the *micro*- and *macroaveraged F-measure* metrics.

Dataset. A random sample of 50 webpage images is manually annotated as *image, text,* or *mixture* by regular internet users of no particular age group. To understand the subjectivity of this task, the interannotator agreement was measured using the Kappa statistic which equates to 0.366. The correct label for the sample images is the one that achieved the most votes.

Table 1: Confusion matrix for the classification task.

	Prediction			
	Text	Mixture	Image	Total
Actual				
Text	6	0	4	10
Mixture	2	4	5	11
Image	4	0	25	29
Total	12	4	34	50

Results. By looking at Table 1, it is immediately observable that the classifier exhibits the worst performance on the *mixture* class by falsely predicting the majority of such samples as *image* instances. This behaviour relates to the way human annotators perceived *mixture* images. Indeed, it has been noted that most of the sample images of the dataset containing even a very small percentage of text were labelled as *mixture* images, creating inconsistencies with the actual meaning of the *mixture* class.

Despite this bad performance on the *mixture* class, the overall F_{macro} measure equates to 60%. This suggests that an acceptable behaviour is demonstrated on the *text* and *image* classes; otherwise this value would have been significantly lower (Table 2).

4.2. HEADINGS_G Inspection Evaluation Experiments

4.2.1. Experiment 2

Aim. To assess the overall performance of the webpage heading identification algorithm for the tasks of detecting the headings of a webpage and assigning heading levels by comparing results against the dataset through *precision* and *recall* for both tasks independently.

Dataset. A set of 313 headings was manually collected from a total of 57 webpages that are unique in the way their headings are styled to be able to test the effectiveness of the algorithm with different headings presentation.

Results. Results for the heading detection task are presented in Table 3, where TP corresponds to the number of correctly identified headings, whilst FP indicates the number of falsely detected headings. A precision of 87% means that the identified headings are almost always real headings of a webpage. A recall of 76% implies that the algorithm is not always capable of detecting all the headings of a webpage. This is, namely, due to the occasional elimination of relevant content when cleaning a webpage from its boilerplate. Nevertheless, these results suggest that, in most cases, the algorithm is capable of correctly identifying the majority of the headings in a webpage.

Table 2: Classification performance evaluation results.

Accuracy	F_{micro}	F_{macro}
70%	70%	60%

Table 3: Evaluation results for the heading detection task.

	Number of headings	TP	FP	Recall	Precision
Total	329	251	39	76%	87%

Table 4: Evaluation results for the heading-level assignment task.

Heading level	Number of headings	TP	FP	Recall	Precision
1	76	55	75	72%	42%
2	200	104	19	52%	85%
3	53	8	0	15%	100%
Total	329	167	94	51%	64%

Table 4 presents results for the heading-level assignment task, where TP is the number of headings correctly assigned a heading level and FP is the number of headings incorrectly assigned a heading level. It can be noted that the precision and recall for this task are significantly lower than those of Table 3.

The main reason for this is again the occasional elimination of relevant content when cleaning a webpage from its boilerplate. Indeed, in such cases, the first detected heading assigned a level of one is most often a child of an eliminated heading. Consequently, higher heading levels are assigned to the subsequent headings when, in reality, they should be given a lower level. This leads to a very low precision for the first heading level. Nonetheless, it has been observed that parent-child heading relationships were correctly identified in most cases (note the high precision for levels 2 and 3). Thus, improving the recall for the headings detection task will also improve the performance of the algorithm for the heading-level assignment task.

4.2.2. Experiment 3

Aim. To measure the degree of correlation between heading-to-content semantic relatedness results computed by the IUE tool through ESA against those manually annotated by human judges captured in the dataset by measuring Pearson's Correlation.

Dataset. A random sample of 50 heading-content pairs was manually annotated by regular internet users of no particular age group using a Likert 5-point scale (Very Bad—0.1; Bad—0.3; Neither Good nor Bad—0.5; Good—0.7; Very Good—0.9). To understand the subjectivity of this task, the interannotator agreement was measured using the weighted Kappa statistic which equates to 0.428. The correct label for the sample heading-content pairs is the one that achieved the most votes.

Results. Results confirm a statistically significant, strong, positive correlation between the values obtained by the IUE tool and those annotated by human judges ($R_{\text{IUE}} = 0.66$, $N = 50$, $P < 0.01$). This correlation is also superior to that obtained when comparing the random baseline with human judgements ($R_{\text{RAND}} = 0.12$). These results can be better visualized from the scatter plots of Figure 3.

Thus, it can be concluded that the IUE tool is strongly capable of modelling human behaviour when determining the semantic relatedness of webpage headings against their content. In turn, this suggests that the IUE tool is suitable for surfacing heading-to-content semantic relatedness violations.

4.2.3. Experiment 4

Aim. To measure the degree of correlation between the heading-to-heading uniqueness scores computed by the IUE tool through the N-gram similarity measure against those manually annotated by human judges captured in the dataset by measuring Pearson's Correlation.

Dataset. A random sample of 50 heading-heading pairs was manually annotated by regular internet users of no particular age group using a Likert 5-point scale (Not Unique—0.1; Not Very Unique—0.3; Moderately Unique—0.5; Unique—0.7; Very Unique—0.9). To understand the subjectivity of this task, the *interannotator agreement* was measured using the weighted Kappa statistic which equates to 0.329. The correct label for the sample heading-heading pairs is the one that achieved the most votes.

Results. Results confirm a statistically significant, moderate, positive correlation between the values obtained by the IUE tool and those annotated by human judges ($R_{\text{IUE}} = 0.57$, $N = 50$, $P < 0.01$). This correlation is also superior to that obtained when comparing the random baseline with human judgements ($R_{\text{RAND}} = -0.006$). These results can be better visualized from the scatter plots of Figure 4.

Thus, it can be concluded that the IUE tool is moderately capable of modelling human behaviour when determining the degree of uniqueness between every pair of webpage headings. In turn, this suggests that the IUE tool is suitable for surfacing heading-to-heading uniqueness violations.

4.3. HOMEPAGE_G Inspection Evaluation Experiments

4.3.1. Experiment 5

Aim. To measure the degree of correlation between the homepage-to-webpage structural layout similarity scores computed by the IUE against those manually annotated by

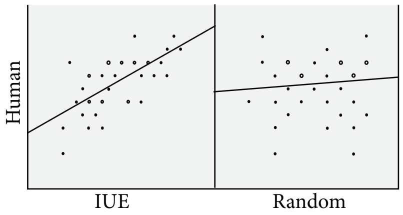


Figure 3: Visualizing the correlation between the IUE tool, random baseline, and human annotators for the task of detecting heading-to-content semantic relatedness.

human judges captured in the dataset by measuring Pearson's Correlation.

Dataset. A random sample of 50 homepage-webpage pairs was manually annotated by regular internet users of no particular age group using a Likert 5-point scale (Very Dissimilar—0.1; Dissimilar—0.3; Neither Similar nor Dissimilar—0.5; Similar—0.7; Very Similar—0.9). To understand the subjectivity of this task, the interannotator agreement was measured using the weighted Kappa statistic which equates to 0.394. The correct label for the sample homepage-webpage pairs is the one that achieved the most votes.

Results. Results confirm a statistically significant, strong, positive correlation between the values obtained by the IUE tool and those annotated by human judges ($R_{\text{IUE}} = 0.77$, $N = 50$, $P < 0.01$). This correlation is also superior to that obtained when comparing the random baseline with human judgements ($R_{\text{RAND}} = .23$). These results can be better visualized from the scatter plots of Figure 5.

Thus, it can be concluded that the IUE tool is strongly capable of modelling human behaviour when determining structural layout similarities between homepages and interior pages of a website. In turn, this suggests that the IUE tool is suitable for detecting homepage-to-webpage structural layout similarity violations.

4.3.2. Experiment 6

Aim. To measure the degree of correlation between the look-and-feel similarity scores computed by the IUE tool against those manually annotated by human judges captured in the dataset by measuring Pearson's Correlation.

Dataset. A random sample of 50 webpage-webpage pairs was manually annotated by regular internet of no particular age group using a Likert 5-point scale (Very Dissimilar—0.1; Dissimilar—0.3; Neither Similar nor Dissimilar—0.5; Similar—0.7; Very Similar—0.9). To understand the subjectivity of this task, the interannotator agreement was measured using the weighted Kappa statistic which equates to 0.368.

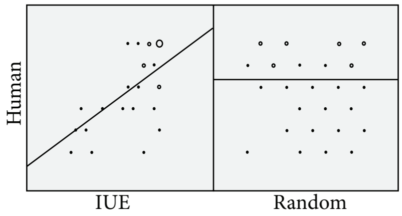


Figure 4: Visualizing the correlation between the IUE tool, random baseline, and human annotators for the task of detecting heading-to-heading uniqueness.

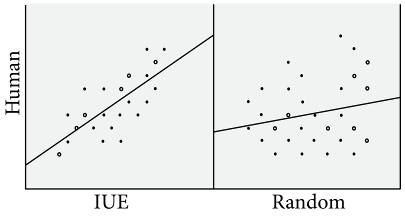


Figure 5: Visualizing the correlation between the IUE tool, random baseline, and human annotators for the task of detecting homepage-to-webpage structural layout similarities.

The correct label for the sample heading-content pairs is the one that achieved the most votes.

Results. Results confirm a statistically significant, strong, positive correlation between the values obtained by the IUE tool and those annotated by human judges ($R_{\text{IUE}} = 0.65$, $N = 50$, $P < 0.01$). This correlation is also superior to that obtained when comparing the random baseline with human judgements ($R_{\text{RAND}} = -0.004$). These results can be better visualized from the scatter plots of Figure 6.

Thus, it can be concluded that the IUE tool is strongly capable of modelling human behaviour when determining look-and-feel similarities between the various pages of a website. In turn, this suggests that the IUE tool is suitable for surfacing look-and-feel inconsistencies among the pages of a website.

5. Conclusions and Future Work

To date, the literature seems to report no solution to the problems of automatically inspecting the guidelines considered for this dissertation with the aim of evaluating websites for usability. To address this challenge, a number of solutions to related problems are adapted suitably and employed to solve the emergent issues when automating the guideline inspection processes. This reflects the novel contribution of this study to the field of automatic website usability evaluation.

The proposed IUE tool is designed and developed to satisfy the requirements specified in Section 2. Experiment results conclude that this tool adequately models human judgement when detecting usability violations, thus justifying its role as a website usability evaluator. Although the main purpose of the IUE tool is to assist website designers, internet

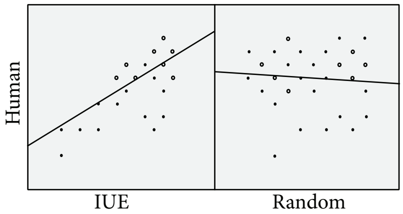


Figure 6: Visualizing the correlation between the IUE tool, random baseline, and human annotators for the task of detecting webpage-to-webpage look-and-feel similarities.

users at large indirectly benefit from its consequences as websites will eventually be created in a more usable manner thus enhancing their online experience. One can further appreciate the significance of this tool after being aware of the importance of usability on the web and the need for its evaluation.

An obvious limitation of the artefact relates to the number of guidelines used for inspection. For the purpose of this dissertation, the current version of the IUE tool references only three out of the 242 guidelines of [31] by virtue of their challenging, nontrivial, and interesting nature. This referencing scheme can be further extended by incorporating additional guidelines. The tool can be enhanced by registering its users to store their website's usability evaluation results. During future evaluations, this stored information can be referenced to demonstrate the improvements or degradations of the website's level of usability.

Additionally, since the tool relies on heavy backend processes which are likely to be time-consuming, it would be ideal to notify the user of the evaluation progress by displaying the estimated time of completion. Moreover, by means of NLP techniques, the current implementation of the webpage heading detection algorithm can be improved so that only text with a valid sentence structure can be classified as a candidate heading.

Possible future work might take this idea of automatic usability evaluation a step further by integrating such a tool directly within website development environments. In this way, usability violations could be surfaced in real time as the website is being created. Such a doing frees users from having to browse for the tool online and thus further facilitates the usability evaluation of a website.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] P. Herbig and B. Hale, "Internet: the marketing challenge of the twentieth century," *Internet Research*, vol. 7, no. 2, pp. 95–100, 1997.

[2] E. B. Kim and S. B. Eom, "Designing effective cyber store user interface," *Industrial Management and Data Systems*, vol. 102, no. 5, pp. 241–251, 2002.

[3] Y. Bart, V. Shankar, F. Sultan, and G. L. Urban, "Are the drivers and role of online trust the same for all web sites and consumers? A large-scale exploratory empirical study," *Journal of Marketing*, vol. 69, no. 4, pp. 133–152, 2005.

[4] A. I. Wasserman, "How the Internet transformed the software industry," *Journal of Internet Services and Applications*, vol. 2, no. 1, pp. 11–22, 2011.

[5] D. Buhalis and B. Neuhofer, "Everything you need to know about internet marketing," *Annals of Tourism Rsearch*, vol. 39, no. 2, pp. 1266–1268, 2012.

[6] S. A. Becker and F. E. Mottay, "A global perspective on web site usability," *IEEE Software*, vol. 18, no. 1, pp. 54–61, 2001.

[7] International Organisation for Standardisation, *ISO9241 Ergonomic, Part 11: Guidance on Usability*, International Organisation for Standardisation, Geneva, Switzerland, 1st edition, 1998.

[8] "Introduction to Web Accessibility," World Wide Web Consortium (W3C), 2005, http://www.w3.org/WAI/intro/accessibility.php.

[9] S. Krug, *Don't Make Me Think: A Common Sense Approach to Web Usability*, New Riders Press, Berkeley, Calif, USA, 2nd edition, 2006.

[10] G. Brajnik, "Automatic web usability evaluation: what needs to be done?" in *Proceedings of the 6th Conference on Human Factors and the Web*, Austin, Tex, USA, 2000.

[11] J. Nielsen, "Usability 101: Introduction to Usability," August 2003, http://www.useit.com/alertbox/20030825.html.

[12] J. Nielsen and H. Loranger, *Prioritizing Web Usability*, New Riders Press, Berkeley, Calif, USA, 2006.

[13] J. J. Cappel and H. Zhenyu, "A usability analysis of company websites," *Journal of Computer Information Systems*, vol. 48, no. 1, pp. 117–123, 2007.

[14] J. Fisher, J. Bentley, R. Turner, and A. Craig, "SME Myths: if we put up a website customers will come to us: why usability is important," in *18th Bled eConference: eIntegration in Action*, Bled, Slovenia, 2005.

[15] B. M. Muir and N. Moray, "Trust in automation, part II: experimental studies of trust and human intervention in a process control simulation," *Ergonomics*, vol. 39, no. 3, pp. 429–460, 1996.

[16] J. S. Rhodes, "How to Gain the Trust of Your Users," September 1998, http://webword.com/moving/trust.html.

[17] C. Flavián, M. Guinalíu, and R. Gurrea, "The role played by perceived usability, satisfaction and consumer trust on website loyalty," *Information and Management*, vol. 43, no. 1, pp. 1–14, 2006.

[18] D. Byun and G. Finnie, "Evaluating usability, user satisfaction and intention to revisit for successful e-government websites," *Electronic Government*, vol. 8, no. 1, pp. 1–19, 2011.

[19] S. Smith and J. Mosier, *MTR-9240 Guidelines for Designing User Interface Software*, Mitre Corporation.

[20] D. Norman, *The Design of Everyday Things*, Doubleday, Broadway, NY, USA, 1988.

[21] J. Nielsen, "The usability engineering life cycle," *Computer*, vol. 25, no. 3, pp. 12–22, 1992.

[22] T. Comber, "Building usabel web pages: an HCI perspective," in *Proceedings of the Australian Conference on the Web (AusWeb '95)*, Ballina, Australia, 1995.

[23] D. Sano, *Desining Large-Scale Websites: Avisual Design Methodology*, Wiley Computer Publishing, John Wiley & Sons, New York, NY, USA, 1996.

[24] J. A. Borges, I. Morales, and N. J. Rodriguez, “Guidelines for designing usable World Wide Web pages,” in *Proceedings of the 1996 Conference on Human Factors in Computing Systems: Common Ground (CHI ’96)*, pp. 277–278, Vancouver, Canada, April 1996.

[25] J. Spool, T. Scanlon, C. Snyder, and T. DeAngelo, *Website Usability: A Designer’s Guide*, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 1998.

[26] J. Fleming, *Web Navigation: Designing the User Experience*, O’Reilly & Associates, Sebastopol, Calif, USA, 1998.

[27] L. Rosenfeld and P. Morville, *Information Architecture for the World Wide Web*, O’Reilly & Associates, Sebastopol, Calif, USA, 1998.

[28] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley, Reading, Mass, USA, 1998.

[29] J. Nielsen, “User interface directions for the web,” *Communications of the ACM*, vol. 42, no. 1, pp. 65–72, 1999.

[30] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*, Pearson Education, Essex, UK, 3rd edition, 2004.

[31] J. Mifsud, *USEFul: a framework to mainstream web site usability through automated evaluation [B.Sc. thesis]*, University of London, 2011.

[32] M. Leavitt and B. Shneiderman, *The Research-Based Web Design & Usability Guidelines*, U.S. Government Printing Office, Washington, DC, USA, 2006.

[33] J. Nielsen and K. Pernice, *Eyetracking Web Usability*, New Riders Press, Berkeley, Calif, USA, 2010.

[34] J. Nielsen and M. Tahir, *Homepage Usability: 50 Websites Deconstructed*, New Riders Press, Berkeley, Calif, USA, 2002.

[35] T. K. Chiew and S. S. Salim, “Webuse: website usability evaluation tool,” *Malaysian Journal of Computer Science*, vol. 16, no. 1, pp. 47–57, 2003.

[36] C. M. Karat, “A comparison of user interface evaluation methods,” in *Usability Inspection Methods*, John Wiley & Sons, New York, NY, USA, 1994.

[37] C. Ardito, P. Buono, D. Caivano et al., “Usability evaluation: a survey of software development organizations,” in *Proceedings of the 23rd International Conference on Software Engineering & Knowledge Egineering*, pp. 282–287, Miami, Fla, USA, July 2011.

[38] C. M. Karat, “Cost-justifying usability engineering in the software life cycle,” in *Handbook of Human-Computer Interaction*, Elsevier, 1997.

[39] S. M. Dray and C. M. Karat, “Human factors cost justification for an internal development project,” in *Cost-Justifying Usability*, Academic Press, 1994.

[40] S. Reed, “Who defines usability? You do!,” *PC Computing*, vol. 5, no. 12, pp. 220–232, 1992.

[41] M. Y. Ivory and M. A. Hearst, “The state of the art in automating usability evaluation of user interfaces,” *ACM Computing Surveys*, vol. 33, no. 4, pp. 470–516, 2001.

[42] A. Beirekdar, M. Keita, M. Noirhomme, F. Randolet, J. Vanderdonckt, and C. Mariage, “Flexible reporting for automated usability and accessibility evaluation of web sites,” in *Human-Computer Interaction—INTERACT, 2005*, pp. 281–294, 2005.

[43] J. Vanderdonckt and A. Beirekdar, “Automated evaluation of web usability by guideline review,” *Journal of Web Engineering*, vol. 4, no. 2, pp. 102–117, 2005.

[44] A. Holzinger, “Usability engineering methods for software developers,” *Communications of the ACM*, vol. 48, no. 1, pp. 71–74, 2005.

[45] R. Otaiza, C. Rusu, and S. Roncagliolo, “Evaluating the usability of transactional web sites,” in *Proceedings of the 3rd International Conference on Advances in Computer-Human Interactions (ACHI ’10)*, pp. 32–37, Saint Maarten, The Netherlands, February 2010.

[46] J. Nielsen, “Usability inspection methods,” in *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 377–378, Boston, Mass, USA, May 1994.

[47] A. Madan and S. K. Dubey, “Usability evaluation methods: a literature review,” *International Journal of Engineering Science and Technology*, vol. 4, no. 2, pp. 590–599, 2012.

[48] T. Conte, J. Massollar, E. Mendes, and G. H. Travassos, “Usability evaluation based on Web design perspectives,” in *1st International Symposium on Empirical Software Engineering and Measurement (ESEM ’07)*, pp. 146–155, Madrid, Spain, September 2007.

[49] M. Y. Ivory and A. Chevalier, *A Study of Automated Web Site Evaluation Tools*, Department of Computer Science, University of Washington, 2002.

[50] R. Benbunan-Fich, “Using protocol analysis to evaluate the usability of a commercial web site,” *Information and Management*, vol. 39, no. 2, pp. 151–163, 2001.

[51] D. A. Bowman, J. L. Gabbard, and D. Hix, “A survey of usability evaluation in virtual environments: classification and comparison of methods,” *Presence*, vol. 11, no. 4, pp. 404–424, 2002.

[52] X. Ferré, N. Juristo, H. Windl, and L. Constantine, “Usability basics for software developers,” *IEEE Software*, vol. 18, no. 1, pp. 22–29, 2001.

[53] H. Desurvire, D. Lawrence, and M. Atwood, “Empiricism versus judgment: comparing user interface evaluation methods on a new telephone-based interface,” *ACM SIGCHI Bulletin*, vol. 23, no. 4, pp. 58–59, 1991.

[54] R. Jeffries, J. R. Miller, C. Wharton, and K. Uyeda, “User interface evaluation in the real world: a comparison of four techniques,” in *Proceedings of the Conference on Human Factors in Computing Systems*, New York, NY, USA, 1991.

[55] L. M. Tobar, P. M. L. Andrés, and E. L. Lapena, “WebA: a tool for the assistance in design and evaluation of websites,” *Journal of Universal Computer Science*, vol. 14, no. 9, pp. 129–139, 2008.

[56] M. B. Skov and J. Stage, “Supporting web developers in evaluating usability and identifying usability problems,” in *Integrating Usability Engineering for Designing the Web Experience: Methodologies and Principles*, pp. 1–14, IGI Global, 2010.

[57] J. Vanderdonckt, “Development milestones towards a tool for working with guidelines,” *Interacting with Computers*, vol. 12, no. 2, pp. 81–118, 1999.

[58] J. O. Bak, K. Nguyen, P. Risgaard, and J. Stage, “Obstacles to usability evaluation in practice: a survey of software development organizations,” in *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges*, pp. 23–32, Lund, Sweden, October 2008.

[59] A. Lecerof and F. Paternò, “Automatic support for usability evaluation,” *IEEE Transactions on Software Engineering*, vol. 24, no. 10, pp. 863–888, 1998.

[60] R. Molich, N. Bevan, S. Butler et al., “Comparative evaluation of usability tests,” in *Proceedings of the 1998 UPA Conference*, Chicago, Ill, USA, 1998.

[61] J. Nielsen, *Usability Engineering*, Academic Press, Boston, Mass, USA, 1993.

[62] M. Matera, F. Rizzo, and G. Toffetti Carughi, “Web usability: principles and evaluation methods,” in *Web Engineering*, pp. 143–148, Springer, 2006.

[63] *MAGENTA—Multi-Analysis of Guidelines by an Enhanced Tool for Accessibility*, The Human-Computer Interaction Group Laboratory, 2010, http://giove.isti.cnr.it/tools/MAGENTA/home.

[64] “Web Content Accessibility Guidelines—WCAG v. 1. 0,” World Wide Web Consortium (W3C), 1999, http://www.w3.org/TR/WCAG10.

[65] “Law n. 4 January 9, 2004—Provisions to support the access to information technologies for the disabled (also known as, “The Stanca act”),” Official Gazette of the Italian Republic, 2004.

[66] R. Atterer, “Model-based automatic usability validation—a tool concept for improving web-based UIs,” in *Proceedings of the 5th Nordic Conference on Human-Computer Interaction (NordiCHI ’08): Buidling Bridges*, pp. 13–22, Lund, Sweden, October 2008.

[67] “Présentation du validateur d’accessibilité OCAWA,” OCAWA, 2010, http://www.ocawa.com/fr/Accueil.htm.

[68] “Référentiel Général d’Accessibilité pourles Administrations—RGAA,” Lé Comité Interministériel du Handicap, 2005.

[69] E. H. Chi, “The Bloodhound Project: automating discovery of web usability issues using the InfoScent simulator,” in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI ’03)*, Ft. Lauderdale, Fla, USA, 2003.

[70] M. Kitajima and P. Polson, “A comprehension-based model of exploration,” *Human Computer Interaction*, vol. 12, no. 4, pp. 345–389, 1997.

[71] C. Lewis, P. Polson, C. Wharton, and J. Rieman, “Testing a walkthrough methodology for theory-based design and walk-up-and-use interfaces,” in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems: Empowering People*, Seattle, Wash, USA, 1990.

[72] P. G. Polson, C. Lewis, J. Rieman, and C. Wharton, “Cognitive walkthroughs: a method for theory-based evaluation of user interfaces,” *International Journal of Man-Machine Studies*, vol. 36, no. 5, pp. 741–773, 1992.

[73] M. H. Blackmon, P. G. Polson, M. Kitajima, and C. Lewis, “Cognitive walkthrough for the Web,” in *Proceedings of the SIGCHI Conference on Huma Factors in Computing Systems: Changing our World, Changing Ourselves*, pp. 463–470, Minneapolis, Minn, USA, April 2002.

[74] G. Branjik, “Using automtic tools in accessibility and usability assurance processes,” in *Proceedings of the 8th ERCIM Workshop on User Interfaces for All*, Lecture Notes in Computer Science, Vienna, Austria, 2004.

[75] E. H. Chi, P. Pirolli, and J. Pitkow, “Scent of a site: a system for analyzing and predicting information scent, usage, and usability of a Web site,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 161–168, The Hague, The Netherlands, April 2000.

[76] M. Y. Ivory, R. R. Sinha, and M. A. Hearst, “Empirically validated web page design metrics,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 53–60, Seattle, Wash, USA, April 2001.

[77] M. Ivory, “Web TANGO: towards automated comparison of informaiton-centric website designs,” in *Proceedings of the ACM Conference on Human Factors in Computing Systems, Student Posters (CHI ’00)*, 2000.

[78] M. Ivory, J. Mankoff, and A. Le, “Using automated tools to improve website usage by users with diverse abilities,” *It & Society*, vol. 1, no. 3, pp. 195–236, 2003.

[79] F. Montero, P. Gonzales, M. Lozano, and J. Vanderdonckt, “Quality modles for automated evaluation of websites usability and accessibility,” in *Proceedings of the International COST294 Workshop on User Interface Quality Model*, Rome, Italy, 2005.

[80] M. Y. Ivory and M. A. Hearst, “Statistical profiles of highly-rated web sites,” in *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 367–374, April 2002.

[81] R. Atterer, A. Schmidt, and H. Hussmann, “Extending web engineering models and tools for automatic usability validation,” *Journal of Web Engineering*, vol. 5, no. 1, pp. 43–64, 2006.

[82] T. Tiedtke, C. Martin, and N. Gerth, “AWUSA—a tool for automated website usability analysis,” in *Proceedings of the 9th International Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS ’02)*, 2002.

[83] M. A. El-Shayeb, S. R. El-Beltagy, and A. Rafea, “Extracting the latent hierarchical structure of web documents,” in *Advanced Internet Based Systems and Applications*, Springer, Berlin, Germany, 2009.

[84] E. Gabrilovich and S. Markovitch, “Computing semantic relatedness using Wikipedia-based explicit semantic analysis,” in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007.

[85] O. Boiman, E. Shechtman, and M. Irani, “In defense of nearest-neighbor based image classification,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’08)*, June 2008.

[86] P. Bohunsky and W. Gatterbauer, “Visual structure-based web page clustering and retrieval,” in *Proceedings of the 19th International Conference on World Wide Web*, pp. 1067–1068, Raleigh, NC, USA, April 2010.

[87] T. Garsiel and P. Irish, “How Browsers Work: Behind the Scenes of Modern Web Browsers,” Google Project, August 2011, http://www.html5rocks.com/en/tutorials/internals/howbrowserswork/.

[88] Y. Takama and N. Mitsuhashi, “Visual similarity comparison for Web page retrieval,” in *Proceedings of the IEEE/WIC/ACM International conference on Web Intelligence (WI’ 05)*, pp. 301–304, September 2005.

[89] M. Kovacevic, M. Diligenti, M. Gori, and V. Milutinovic, “Visual adjacency multigraphs—a novel approach for a web page classification,” in *Proceedings of the SAWM04 Workshop (ECML ’04)*, 2004.

[90] X. Qi and B. D. Davison, “Web page classification: features and algorithms,” *ACM Computing Surveys*, vol. 41, no. 2, pp. 1–31, 2009.

[91] M. T. Law, C. S. Gutierrez, N. Thome, and S. Gançarski, “Structural and visual similarity learning for web page archiving,” in *Proceedings of the 10th International Workshop on Content-Based Multimedia Indexing (CBMI ’12)*, Annecy, France, 2012.

[92] M. Alpuente and D. Romero, “A visual technique for web pages comparison,” *Electronic Notes in Theoretical Computer Science*, vol. 235, pp. 3–18, 2009.

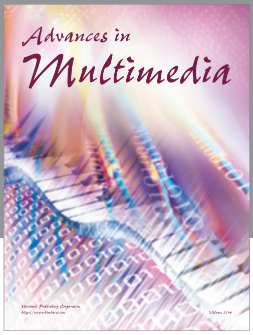
[93] A. Pnueli, R. Bergman, S. Schein, and O. Barkol, *Web Page Layout Via Visual Segmentation*, HP Laboratories, 2009.

[94] J. Cao, B. Mao, and J. Luo, “A segmentation method for web page analysis using shrinking and dividing,” *International Journal of Parallel, Emergent and Distributed Systems*, vol. 25, no. 2, pp. 93–104, 2010.

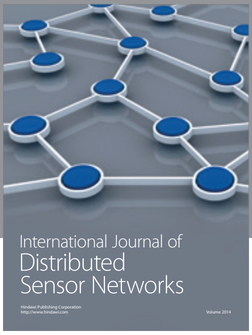
[95] K. Riesen, M. Neuhaus, and H. Bunke, “Bipartite graph matching for computing the edit distance of graphs,” in *Graph-Based Representations in Pattern Recognition*, pp. 1–12, Springer, 2007.

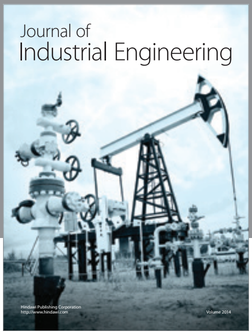
[96] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (SecureComm '08)*, September 2008.

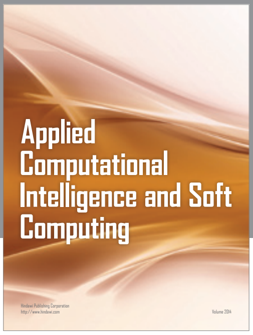
[97] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, Istanbul, Turkey, September 2008.

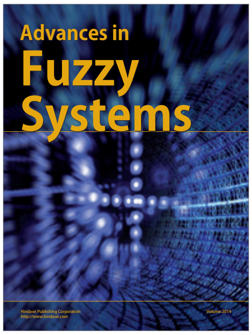

Advances in
Multimedia

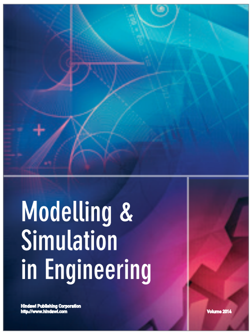
The Scientific
World Journal

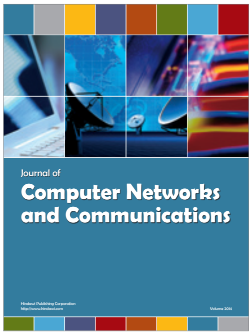

International Journal of
Distributed
Sensor Networks

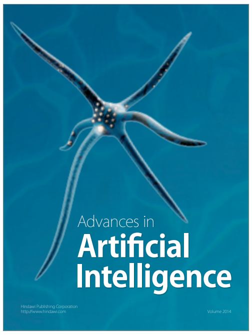

Journal of
Industrial Engineering

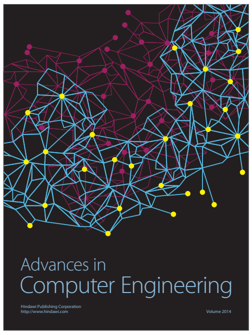

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

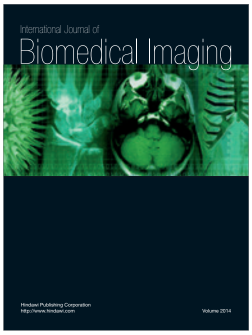

Modelling &
Simulation
in Engineering

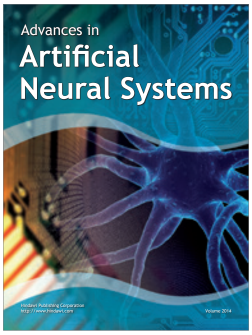

Journal of
Computer Networks
and Communications

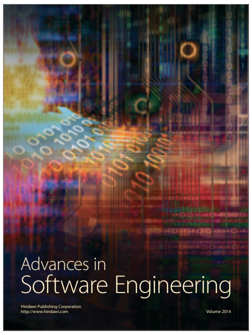

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

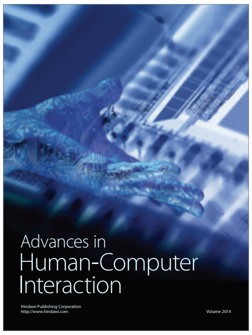
International Journal of
Computer Games
Technology

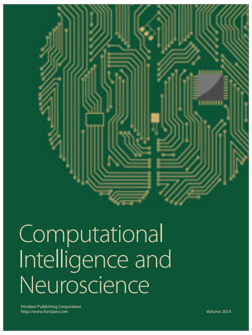

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering