Flexible Reporting for Automated Usability and Accessibility Evaluation of Web Sites

Abdo Beirekdar[1], Marc Keita[1], Monique Noirhomme[1], Frédéric Randolet[1], Jean Vanderdonckt[2], and Céline Mariage[2]

[1] Fac. Univ. Notre-Dame de la Paix, Institut d'Informatique, Rue Grandgagnage, 21
B-5000 Namur (Belgium)
{abe, mno, mke, fra}@info.fundp.ac.be
http://www.info.fundp.ac.be

[2] Université catholique de Louvain, Information Systems Unit, Place des Doyens, 1
B-1348 Louvain-la-Neuve (Belgium)
{vanderdonckt, mariage}@isys.ucl.ac.be
http://www.isys.ucl.ac.be/bchi

Abstract. A system for automatically evaluating the usability and accessibility of web sites by checking their HTML code against guidelines has been developed. All usability and accessibility guidelines are formally expressed in a XML-compliant specification language called Guideline Definition Language (GDL) so as to separate the evaluation engine from the evaluation logics (the guidelines). This separation enables managing guidelines (i.e., create, retrieve, update, and delete) without affecting the code of the evaluation engine. The evaluation engine is coupled to a reporting system that automatically generates one or many evaluation reports in a flexible way: adaptation for screen reading or for a printed report, sorting by page, by object, by guideline, by priority, or by severity of the detected problems. This paper focuses on the reporting system.

1 Introduction

Since the communication and the information transfer are nowadays predominantly achieved through the World Wide Web, the Web probably represents one of the most largely used channels for information exchange [8]. This observation does not necessarily lead to the conclusion that this channel is appropriately tailored for the wide diversity of users, computing platforms, and environments in which users are working, thus provoking a digital divide [13]. In order to reduce this digital divide, the e-Europe action plan (http://europa.eu.int/information_society/eeurope/action_ plan/ eaccess/index_en.htm), accepted by European Countries in June 2000, foresees that any public site should be made compatible with Web Accessibility Initiative (WAI) recommendations recommended by the W3C. The resolution e-Europe 2002-Public Web Site accessibility and their contents (P5-TAPROV-2002-0325) are very precise on this subject: companies in charge of developing web sites for any public administration will be forced to develop sites adhering to these recommendations.

Among others, Usability and Accessibility (U&A) guidelines have been set up to help designers in the process of creating usable and accessible sites. For instance, some organizations like W3C consortium promote recommendations for accessible Web Sites: the Web Accessibility Initiative (WAI) recommendations and the Web Content Accessibility Guidelines [21]. But few designers know the existence of these guidelines. When they are aware of their existence, they are confronted with several problems [20]: too many guidelines, conflicting guidelines, various interpretations of these guidelines. When designers and developers are still decided to apply and check such guidelines despite their shortcomings, they do not always have the resources required to conduct this process thoroughly and successfully. To address these needs, several tools have been developed with the hope that by using the tool, the resources required for applying and checking guidelines will be decreased, especially the time will be reduced, while still reaching the target of U&A assessment [20].

Automated evaluation of web sites [12] not only represents a tentative to address both the needs of U&A and the requirements of designers and developers, but also a largely underexplored area [10] that could demonstrate promising results [11], but also reveal several shortcomings [3,5,13]. One of these shortcomings, but not the only one, is the capability of the tool to deliver relevant information after U&A evaluation so that the designers and developers could effectively and efficiently improve the existing version of the web site. Without such formative evaluation, it is likely that the results of the evaluation process will remain without the desired impact [11].

In the context of the DESTINE project (Design & Evaluation STudio for INtent-based Ergonomic web sites – www.info.fundp.ac.be/DESTINE), we have developed a system for automating U&A evaluation of web sites. The system implements a novel approach that we developed for automating the evaluation of a web site against U&A guidelines by checking a formal representation of these guidelines on the web pages of interest [4]. The aim of the approach is to overcome the major shortcomings of existing tools [5], mainly the fact that the evaluation logic (the guidelines to be evaluated) are completely embedded and hard coded in the software [4]. The main characteristic of our approach is the separation between the evaluation logic (i.e. the guidelines to be evaluated) and the evaluation engine (i.e. the engine that performs the evaluation of the guidelines). In this way, the U&A guidelines can be expressed in terms of conditions to be satisfied on HTML elements (i.e., tags, attributes). A formal specification language supporting this approach implements a framework [2] that enables the transformation of such U&A guidelines from their initial expression in natural language into testable conditions on the HTML code. Once expressed, the guidelines can be evaluated at evaluation-time by configuring their formal expression in an improved way depending on the guidelines to be evaluated and the HTML elements contained in the page. This process consequently involves the guidelines that are considered relevant to the targeted evaluation context, and factors out sub-structures that are common across these guidelines, even if they come from different sets of guidelines. The results of automatic evaluation are presented in a report. A detailed description of the evaluation process and its fundamental concepts are described in [4,20]. Therefore, this paper will focus on the flexible reporting system that is coupled to the engine.

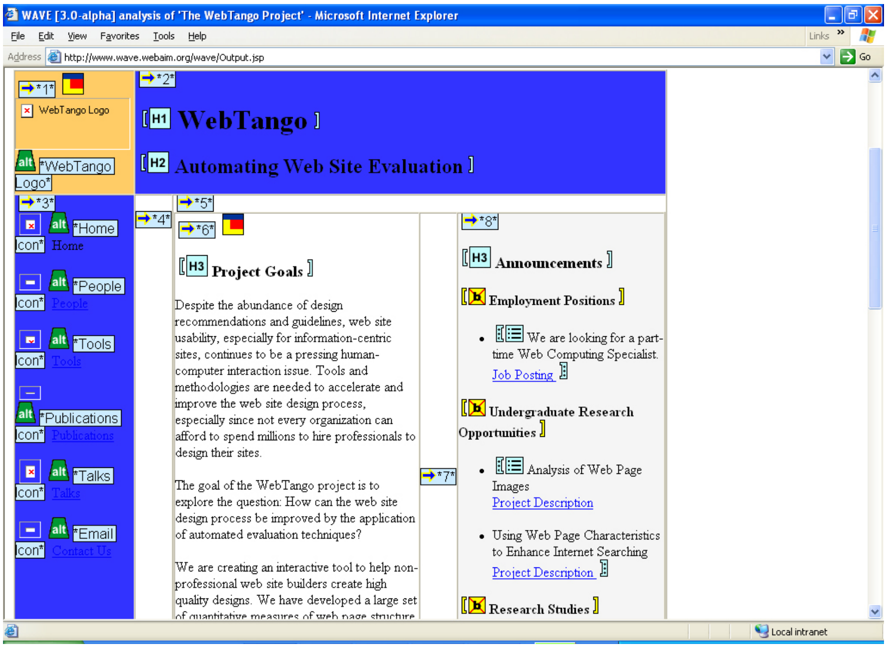
This paper is structured as follows: Section 2 briefly describes some automatic U&A evaluation tools and some examples of generated evaluation reports. Section 3 presents a brief description of the environment. Section 4 explains how different evaluation reports can be generated with different goals in mind. Section 5 concludes the paper by stressing major advantages and the remaining work to be done.

2 Related Work

The general process of performing an automated evaluation of a web site could be decomposed into a sequence of four main steps, that are partially, totally or not at all supported in existing tools for automated evaluation [12]:

1. **Step 1:** Collecting U&A data with their corresponding metrics, such as task completion time, errors, guideline violations, and subjective ratings. A typical manifestation of this step in existing tools consists of conducting a static analysis of the HTML code to ensure that it conforms to U&A guidelines, such as the Section 508 guidelines (US Federal standard) [18], the W3C Content Accessibility guidelines [21], or both. There is a lot of similarity between these two sets of guidelines, because the Section 508 guidelines are based on the W3C guidelines. For this purpose, existing tools collect usage data such as in RemUsine [11], manipulate a task model [11], identify instances of web page components (e.g., widgets, text, graphics, images, fonts) such as in WebTango [12], Kwaresmi [4], A-Prompt [2] so as to perform the U&A analysis. One major shortcoming of this step is that most objects to be evaluated are predetermined according to the evaluation method. It is rarely possible to expand the scope of the existing collecting step.
2. **Step 2:** Analyzing collected U&A data to detect U&A problems in the web site. Existing tools typically attempt to detect deviations between reference values (e.g., a value considered as linked to U&A) and collected values (e.g., the values of the metrics computed in the previous step). Similarly to the previous step, a common shortcoming is that the checkpoints to be evaluated are opportunistically programmed in the software, with little or no possibility to adapt these contents. Bobby (http://watchire.bobby .com), A-Prompt [2], AccessEnable [6], Listener [7], Section 508 verifier [18] cannot handle U&A guidelines other than the one initially implemented (WAI and Section 508). For this reason, a new generation of tools clearly separates the evaluation logic (e.g., the guidelines) from the evaluation engine. These tools typically express guidelines in a structured way, according to a XML-compliant format, that are further parsed and processed on the web pages.
3. **Step 3:** Reporting analysis results to the end user (e.g., web site designer, owner, and visitor). Most tools for automated evaluation, such as Web Static Analyzer Tool (WebSAT) [6], and WebTango [7], report guideline violations in various ways: textually, graphically, numerically, or in some combination. For instance, WAVE [17] produces as output a new web page containing icons added closely to every deviation detected (Fig. 1). Ocawa (http://www.ocawa.com/) displays an accessibility audit report consisting of a series of links leading to individual problems or multiple instances of the same problem type. Ivory [12] observes that the report produced by such tools still demands considerable efforts to interpret the results. For example, WAVE icons are numerous and unintuitive, making their use and interpretation very difficult. Another important observation is that the report format does not vary according to the target user: an evaluator is not the same as a user.
4. **Step 4:** Suggesting solutions or improvements to repair the previously detected problems. The critique tools, such as 508 Accessibility Suite [18], A-Prompt [17], Bobby (http://watchfire.bobby.com), LIFT-NNG, and WAVE [16], also provide recommendations or assistance in repairing violations ([12] provides a detailed dis-

cussion of most of these tools). It is also difficult to explore results or to male repairs with A-Prompt, and LIFT-NNG because each tool presents a list of terse violations within a small window (Fig. 2). The main problems come from the number of structure of a report's page (too many panel in one page increase confusion), its length (difficulty to search information in that kind of page) and the way the errors are identified (many icons are not intuitive to understand).


WAVE [3.0-alpha] analysis of 'The WebTango Project' - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://www.wave.webaim.org/wave/Output.jsp

1 WebTango Logo alt *WebTango Logo*

2 [H1 WebTango]

[H2 Automating Web Site Evaluation]

3 alt *Home Icon* Home

alt *People Icon* People

alt *Tools Icon* Tools

alt *Publications Icon* Publications

alt *Talks Icon* Talks

alt *Email Icon* Contact Us

4 *5* *6*

[H3 Project Goals]

Despite the abundance of design recommendations and guidelines, web site usability, especially for information-centric sites, continues to be a pressing human-computer interaction issue. Tools and methodologies are needed to accelerate and improve the web site design process, especially since not every organization can afford to spend millions to hire professionals to design their sites.

The goal of the WebTango project is to explore the question: How can the web site design process be improved by the application of automated evaluation techniques?

We are creating an interactive tool to help non-professional web site builders create high quality designs. We have developed a large set of quantitative measures of web page structure

7 *8*

[H3 Announcements]

[Employment Positions]

- We are looking for a part-time Web Computing Specialist. Job Posting

[Undergraduate Research Opportunities]

- Analysis of Web Page Images Project Description
- Using Web Page Characteristics to Enhance Internet Searching Project Description

[Research Studies]

Local intranet

Fig. 1. Evaluation report produced by WAVE, where guideline violations are overlaid on the actual web page by using icons to flag potential issues and also depicts the page reading order (arrows with numbers)

As much effort has already been devoted to covering the scope of steps 1 and 2 (e.g., [4,19]), the remaining of this paper will focus on step 3. For this purpose, we will present a new way of reporting evaluation results generated by our automated evaluation tool by showing that a benefit of the evaluation engine is that not only the evaluation could be automated to some extent, but also that the results issued by this engine could be parameterized so as to produce an evaluation report targeted at different types of users. First, the next section will start by providing a brief description of the DESTINE environment itself. Then, in the forthcoming section, the step of producing flexible reports from the evaluation process will be examined in details.

3 DESTINE Evaluation Tool

The goal of DESTINE [13] is to assist any party interested in evaluating the ergonomic quality (mainly, U&A) of web sites based on existing guidelines gathered in guideline bases (e.g., W3C, Section508, custom guidelines). Interested parties include the end user (i.e., the visitor of the web site to know whether the site could be accessed), the designers and the developers (e.g., to know what they can improve in the web site design), or evaluators (i.e., persons who are in charge of assessing the U&A quality of an existing web site, for information purposes or in order to receive official accreditation or certification). It is based on a formal Guideline Definition Language (GDL) to express and structure formally ergonomic recommendations to guarantee the interoperability of the tool (and is a response to the first problem presented in the beginning of this paper). GDL is also compatible with XML, so the user can use different recommendations base in the same tool. DESTINE is open and does not require any existing development environment. Fig. 2 graphically depicts the global architecture of the system. It will be integrated into a Web design environment (e.g., Macromind DreamWeaver) to maximize its access by a web designer or a web developer. The modules of this software architecture are further detailed in the next subsections.

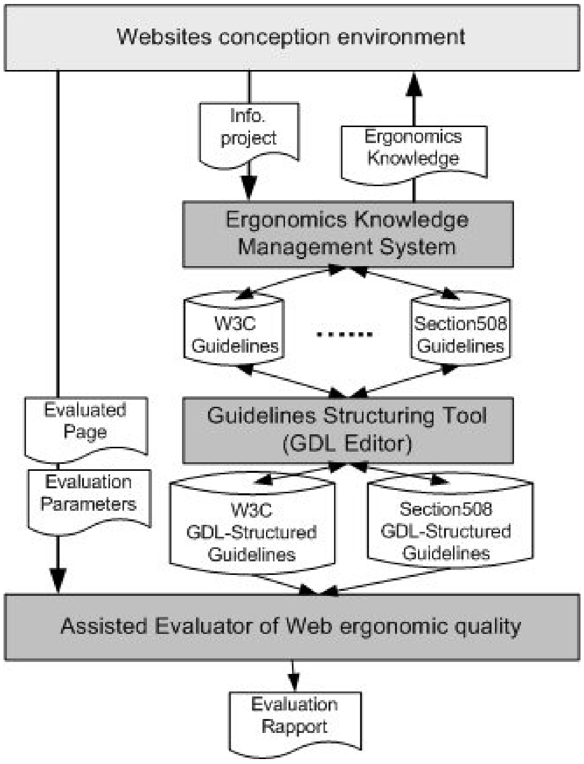

Websites conception environment
Info. project
Ergonomics Knowledge
Ergonomics Knowledge Management System
W3C Guidelines
......
Section508 Guidelines
Evaluated Page
Guidelines Structuring Tool (GDL Editor)
Evaluation Parameters
W3C GDL-Structured Guidelines
Section508 GDL-Structured Guidelines
Assisted Evaluator of Web ergonomic quality
Evaluation Rapport

Fig. 2. Global architecture of the DESTINE system

3.1 The Ergonomic Knowledge Management System

This module manages the ergonomic knowledge contained in guidelines bases during the various steps of the life cycle: creating a new base of guidelines (e.g., WCAG, Section508, etc.), inserting new guidelines in this base, distributed and collaborative editing of the existing guidelines (e.g., it is possible to enrich the base by anyone via a Web browser), selecting the guidelines corresponding to a given context (e.g., targeted user stereotypes, type of site, types of tasks, etc.). In addition to managing the information related to U&A guidelines (e.g., the source, the indexing keys, the comments), one particular field contains the guideline specification in a GDL-compliant form that will be parsed afterwards at evaluation-time.

3.2 GDL Editor

This module is used to formally specify a guideline in a GDL-compliant form and to store it in the guidelines base or in a XML file to be exploited later on by the evaluation engine. To exemplify how a guideline initially expressed in natural language is progressively transformed into a formal interpretation, a simple example of a usability guideline "A page must not have more than 8 links" (fictive usability guideline). As the specification of this guideline progresses, more and more tags are added to provide various levels of description of the intended guideline. First, the guideline is assigned to an ID (here, "Test_G1") and its statement in natural language is provided. Perhaps the guideline can be reproduced here exactly in the same way as it is provided by the usability source. Or perhaps a reformulation of the initial guideline according to a special scheme could be preferred. Since a same guideline could lead to different interpretations on how to apply the guideline at design-time and how to assess it at evaluation-time, each original guideline can be attached to one or many interpreted guidelines. In this way, it is possible to evaluate different interpretations of the same guideline, but depending on the context of use. Then, the evaluation structure specify which HTML tags will be used for the evaluation of this guideline. Several tags could be involved. Therefore, they are gathered in evaluation sets so that different evaluation sets could be considered sequentially or concurrently.

Original Guideline

```
<GDL_Specification>
  <Original_Guideline Name="Test_G1"
      EAspect="Usability" Source="Custom"
      Statement="a page must not have more than 8 links"/>
```
Interpreted Guideline
```
  <Interpretation Context="Test">
    <Interpreted_Guideline Name="Max 8 links"
     Statement="Verify that number of text links is less that
     9"/>
```
HTML Elements (Evaluation structure)
```
  <Evaluation_Structure>
  <HTML_Elements>
    <HTML_Element ID="E1" Tag="A" Attribute="href"/>
  </HTML_Elements>
  <Evaluation_Sets>
     <Evaluation_Set S_ID="S1" Name="Links"
         Description="Check all text links"
         Priority="A">
        <Set_Element Id="E1" Mandatory="true">
           <Scope type="Page"/>
        </Set_Element>
```

```
      </Evaluation_Set>
    </Evaluation_Sets>
  </Evaluation_Structure>
```

Evaluation logic

```
  <Evaluation_Logic>
     <Basic_Values>
        <Basic_Value V_ID="MaxLinkNumber"
            Value="8" Type="Integer"/>
        </Basic_Values>
     <Evaluation Set_ID="S1">
     <Vars>
       <Var Name="set" Type="Set"/>
     </Vars>
     <Operation Op_ID="Op1" Symbol="NumberOfInstances"
            Return_Type="Integer">
       <Argument Type="Var" Value="set"/>
       <Action Result="ANY" What="Jump" Where="Op2"/>
     </Operation>
     <Operation Op_ID="Op2" Symbol="less"
               Return_Type="java.lang.Boolean">
       <Argument Type="Op" Value="Op1"/>
       <Argument Type="Val" Value="MaxLinkNumber"/>
       <Action Result="True" What="Stop"/>
       <Action Result="False" What="Error"
            Why="This page contains more than 8 links."/>
     </Operation>
    </Evaluation>
   </Evaluation_Logic>
</GDL_Specification>
```

Then comes the most important section of a GDL-expressed guideline: the evaluation logic consists of the full declarative definition of the checkpoints to be processed and the actions that need to be taken when deviation with respect to any checkpoint is detected. Briefly said, the above specification provides the following information:

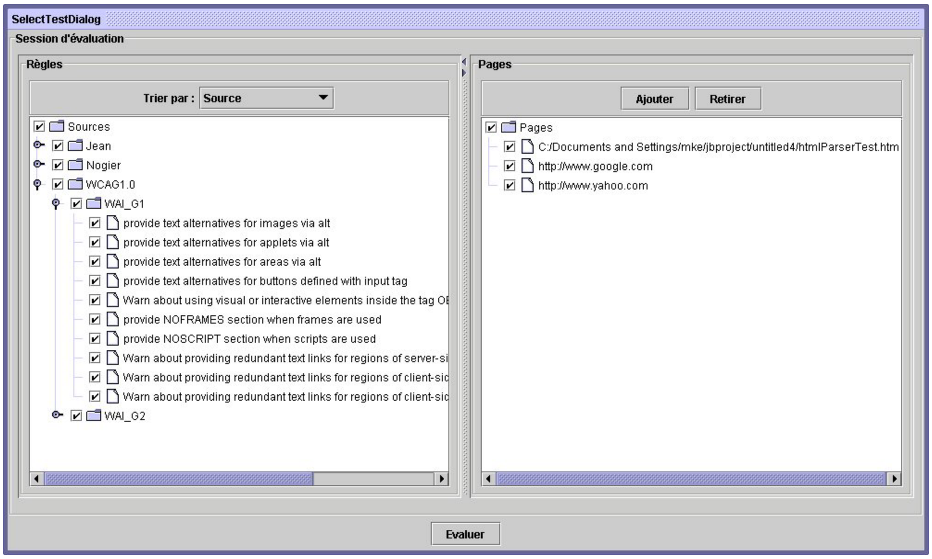
- Guideline statement, related ergonomic aspect, source, etc.
- Interpretation of the original guideline: context, re-expression of the guideline.
- What HTML elements we must examine in the web page to review the guideline, and where to search them (scope of a set element).
- What logic to apply on captured data in order to verify the respect or violation of any guideline.
- What message to send to the users in case of error.

3.3 The GDL Evaluator

On the basis of some evaluation parameters, this module evaluates the ergonomic quality of a page, a series of pages or a whole site by subjecting it to a set of ergonomic guidelines taken from the databases or XML files. It produces a customizable evaluation report. The pages having ergonomic problems are isolated to be treated by the ergonomic reparation tool. We cannot obviously automate the evaluation of all the guidelines in a complete way (the formal GDL specification provides necessary information indicating their level of automation: partial, total, with a percentage).

4 Reporting of Evaluation Results

After specifying the formal guideline, the evaluation module (Fig. 3) can evaluate any web page against it by parsing the conditions that are involved in each checkpoint, interpreting each condition on each instance of objects contained in a web page. All what is needed is to load the formal representation of the guideline (its GD specification) and to provide the URL of the page to be evaluated. This last step could be performed locally (by evaluating a web page or a series of web pages that have been saved from their original web site – *off-line evaluation* – or by evaluating dynamically a web page or a series of web page whose starting URL is provided along with a depth level – *on-line evaluation*).


SelectTestDialog
Session d'évaluation

Règles
Trier par : Source
- ☑ Sources
  - ☑ Jean
  - ☑ Nogier
  - ☑ WCAG1.0
    - ☑ WAI_G1
      - ☑ provide text alternatives for images via alt
      - ☑ provide text alternatives for applets via alt
      - ☑ provide text alternatives for areas via alt
      - ☑ provide text alternatives for buttons defined with input tag
      - ☑ Warn about using visual or interactive elements inside the tag OE
      - ☑ provide NOFRAMES section when frames are used
      - ☑ provide NOSCRIPT section when scripts are used
      - ☑ Warn about providing redundant text links for regions of server-si
      - ☑ Warn about providing redundant text links for regions of client-sic
      - ☑ Warn about providing redundant text links for regions of client-sic
    - ☑ WAI_G2

Pages
Ajouter Retirer
- ☑ Pages
  - ☑ C:/Documents and Settings/mke/jbproject/untitled4/htmlParserTest.htm
  - ☑ http://www.google.com
  - ☑ http://www.yahoo.com

Evaluer

Fig. 3. DESTINE evaluation module

In Fig. 3, the left panel shows a hierarchy of all potential sources of U&A knowledge (e.g., usability guides, style guides, and standards). Each source can be opened to reveal its own table of contents with link to their guidelines. Each section in the table of contents can be selected individually and recursively: any selected entry in the global hierarchy automatically selects all its children (source, section, subsection, guidelines) and vice versa. The design can then select or unselect the evaluation of any guideline depending on the requirements imposed by the evaluation procedure. In this way, the evaluation can be made *on-demand* and can only focus on those guidelines which are considered relevant for this web page. As opposed to a "all or nothing" rule where all guidelines are involved or none. Although a first selection can already be made at this stage, the evaluation engine can also detect guidelines that do not need to be processed depending on the contents of a web page. For instance, if a guideline is assumed to check some properties of a push button, but that the web page of concern does not contain any such push button, the guideline, even if selected, will be left out. The right panel of Fig. 3 gathers in a list all the web pages to be evaluated simultaneously: on-line and/or off-line. As a result of the evaluation, the tool generates a "*dynamic mini-site*" (Fig. 4). The term "mini-site" comes from the composition in a set of HTML pages and the term "dynamic" from the ability of the user in modifying the navigation and the content of the site after the generation of the document (by using "JavaScript" for example). In this way, it is expected that the format of the resulting report could be made adapted to the user profile.

4.1 User Profile

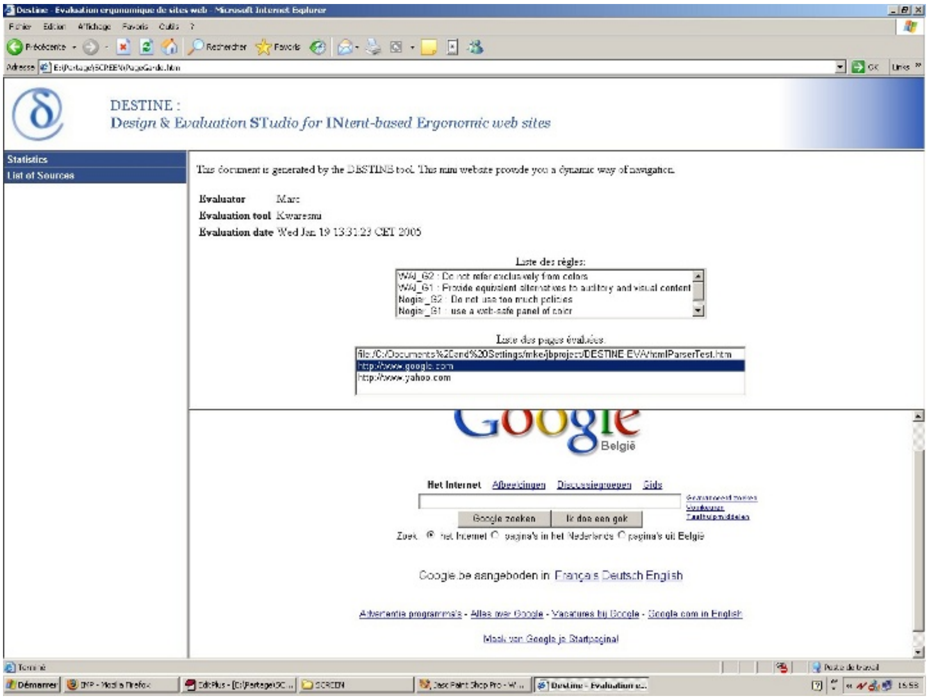
Even if the report has a set of options to modify the presentation of its content, some parameters like the user profile could be specified prior to the report generation. Two different types of user profiles are supported: *expert profile* and *designer profile*. The expert profile is aimed at a human factors expert who does not want to be bothered by the HTML code and who only wants to know information like which guidelines where violated, the seriousness of the errors or their proportions. The *designer profile* will be chosen by a user like a web site designer who needs to know where the errors are located in the HTML code and how to correct them.

4.2 Generated Report

The report, generated for these two profiles, is relatively different to satisfy the needs of the two kinds of user, even if some information will be the same in both versions. The difference between the formats is mainly motivated by to the designer's desire to fix the code depending on the evaluation results, as provided in the report that is automatically generated based on the parameters. For instance, the report presentation may try to optimize its format for printing or for visualization/navigation purposes. In general, evaluation reports produced by other tools are composed by only one block of results, displaying a lot of elements in only one page making it very long to browse. This may prevent the users from viewing the results in an effective way because extracting a clear structure from those heaps of information is not easy.

To obtain a usable navigation, several small and structured HTML pages are produced. The report consists of three main parts: the *menu* (left pane in Fig. 4), the *main frame* (top right pane in Fig. 4), and the page viewer (bottom right pane in Fig. 4) which simply views the evaluated page. The page viewer helps the user to keep an eye on which part of an evaluated page she is working and see what is wrong.

The menu. It is dynamic and can be modified according to the preferences of the user. For example, the guidelines can be sorted by "Source" (as W3C, Section 508), by "Ergonomic Aspect" (as Usability, Accessibility), by "HTML Object" (as tables, images) just by choosing an item in a combination box. In this way, the generated report can accommodate the many variations that may exist between the different potential users of DESTINE. The menu is displayed with the assistance of JavaScript to make it more usable. Even if the browser does not support JavaScript, the menu can still be used without loosing information: all of the menu elements are then shown like a list.


Destine - Evaluation ergonomique de sites web - Microsoft Internet Explorer
Adresse E:\Partage\SCREEN\PageGarde.htm
Statistics
List of Sources
Google België
Het Internet Afbeeldingen Discussiegroepen Gids
Google zoeken
Ik doe een gok
Geavanceerd zoeken
Voorkeuren
Taalhulpmiddelen

DESTINE :
Design & Evaluation STudio for INtent-based Ergonomic web sites

This document is generated by the DESTINE tool. This mini website provide you a dynamic way of navigation.

Evaluator	Marc
Evaluation tool	Kwaresmi
Evaluation date	Wed Jan 19 13:31:23 CET 2005

Liste des règles:

WAI_G2 : Do not refer exclusively from colors
WAI_G1 : Provide equivalent alternatives to auditory and visual content
Nogier_G2 : Do not use too much policies
Nogier_G1 : use a web-safe panel of color

Liste des pages évaluées.

file:/C:/Documents%20and%20Settings/mkej/bproject/DESTINE-EVA/htmlParserTest.htm
http://www.google.com
http://www.yahoo.com

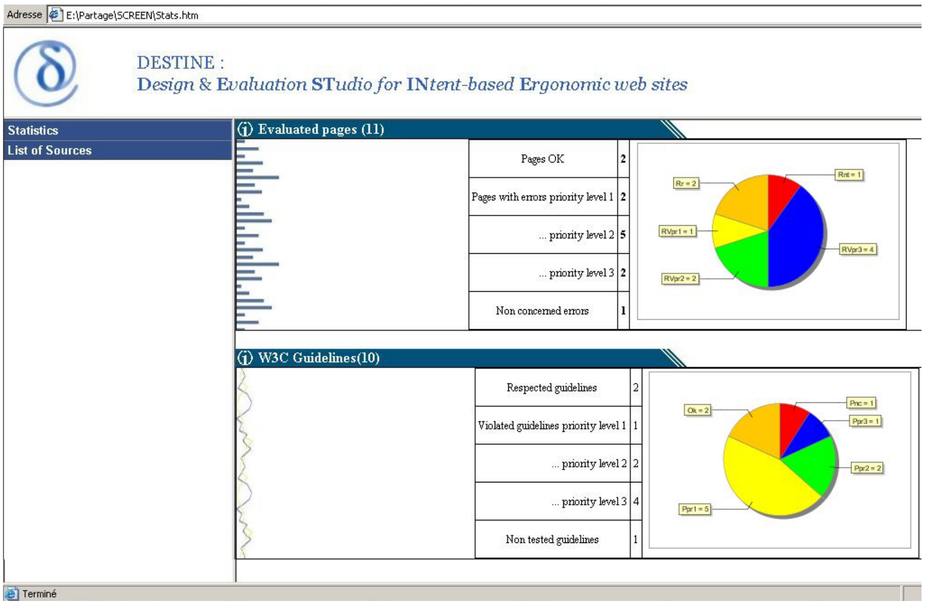
Zoek: het Internet pagina's in het Nederlands pagina's uit België

Google.be aangeboden in: Français Deutsch English

Advertentie programma's - Alles over Google - Vacatures bij Google - Google.com in English

Maak van Google je Startpagina!

Fig. 4. Generated evaluation report



DESTINE :
Design & Evaluation STudio for INtent-based Ergonomic web sites

Evaluated pages (11)

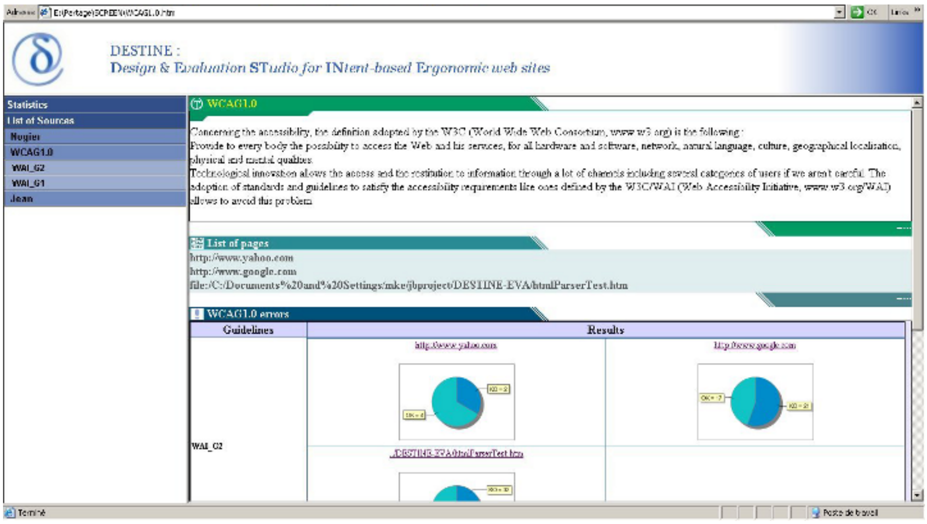
Pages OK	2
Pages with errors priority level 1	2
... priority level 2	5
... priority level 3	2
Non concerned errors	1

W3C Guidelines(10)

Respected guidelines	2
Violated guidelines priority level 1	1
... priority level 2	2
... priority level 3	4
Non tested guidelines	1

Fig. 5. Page of statistics

The main frame. It contains three different page types. The first page type is linked to the global statistics of the evaluation, the second page type is attached to the selected sorting criteria and the last one, to the evaluated page itself. The first page type contains statistical information like the proportion of pages that have passed successfully all checkpoints of priority 1. Another example of statistical information can be the number of pages that have passed through the entire test successfully for a single guideline source. This page also contains graphics providing a global view of the evaluation helping user seeing which criteria is the most/less respected (Fig. 5). The second type of page contains information about the results corresponding to the criteria for all evaluated pages. E.g., those pages can show for each evaluated page some local statistical graphics and theory about the selected criteria, like shown in Fig. 6. If the user selects the first guideline of the W3C level, the page will show information about the theory, graphics of each page compared to the checkpoints of that guideline.


E:\Partage\SCREEN\WCAG1.0.htm

DESTINE :
Design & Evaluation STudio for INtent-based Ergonomic web sites

Statistics
List of Sources
Nogier
WCAG1.0
WAI_G2
WAI_G1
Jean

WCAG1.0

Concerning the accessibility, the definition adopted by the W3C (World Wide Web Consortium, www.w3.org) is the following :
Provide to every body the possibility to access the Web and his services, for all hardware and software, network, natural language, culture, geographical localisation, physical and mental qualities.
Technological innovation allows the access and the restitution to information through a lot of channels including several categories of users if we aren't careful. The adoption of standards and guidelines to satisfy the accessibility requirements like ones defined by the W3C/WAI (Web Accessibility Initiative, www.w3.org/WAI) allows to avoid this problem.

List of pages
http://www.yahoo.com
http://www.google.com
file:/C:/Documents%20and%20Settings/mke/jbproject/DESTINE-EVA/htmlParserTest.htm

WCAG1.0 errors

Guidelines	Results	
WAI_G2	http://www.yahoo.com	http://www.google.com
	../DESTINE-EVA/htmlParserTest.htm	

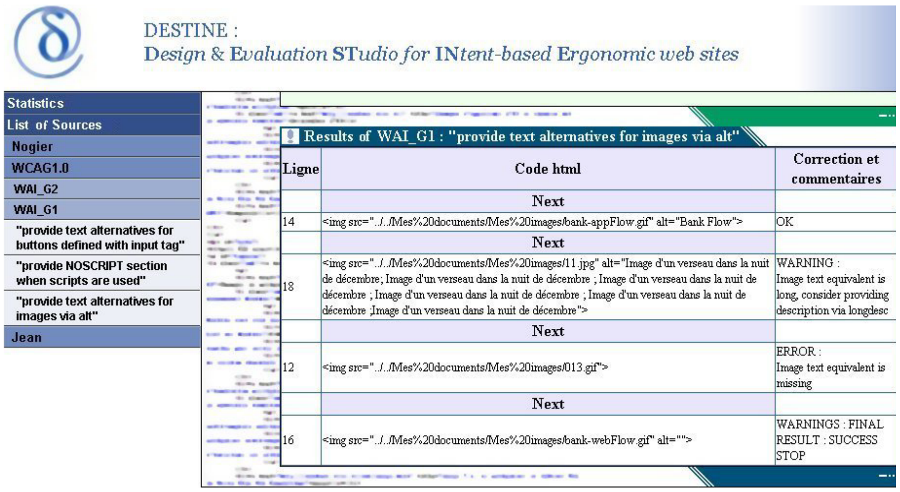
Fig. 6. Statistics by source of guidelines

The last page owns the same type of content as the ones previously found but is focused on one evaluated page at a time. This is the place where the users would retrieve such information as the list of wrong instances and its location in the HTML code, as shown in Fig. 7.

The report as generated by DESTINE therefore presents the following advantages:

1. The navigation within the evaluation results is much more flexible than in existing tools since many navigations correspond to various evaluation intellectual paths, an essential aspect for that kind of document. Information can be found easily (the document is very structured) and in an intuitive way (most of people knows how to navigate through a website) just by clicking on a link and not by scanning a long document. A 'focus+details' navigation scheme can be adopted.

2. The document is structured in several levels of details such as a guideline, a checkpoint, a U&A criteria, a page widget,…. Each level has its own set of data like statistics, theory and comments directly related with it.
3. The visual aspect of the report can be customized by the user, making it more user-friendly. Parameters like the colors, the type of the graphics and the font could be chosen by the user, thus enabling a personalization of the contents.
4. The type of the document (HTML) can be easily read on all platforms without specific software, just by using a browser which is usually included in most of operating systems. In addition, the internal representation of the evaluation report is made partially compatible with EARL V1.0, a W3C recommendation to uniformly present the results of evaluation across tools. In addition, the HTML code generated for the mini-site is itself made compliant with U&A guidelines as this report in itself consists of a web site.
5. The presentation is compliant with accessibility guidelines. We give hereafter some examples. Instead of using frames , which do not guarantee accessibility, we use the style sheets (CSS). The CSS make it possible to simulate frames but to preserve accessibility. All the images have alternate texts describing themselves, thus making it possible to a textual navigator to read them. The tables are also integrated to be comprehensible with such navigators. Finally and always by way of example, the colors and the police can be adapted according to user's needs. The report was tested on textual navigators such as Lynx (http://lynx.browser.org/).


DESTINE :
Design & Evaluation STudio for INtent-based Ergonomic web sites
Statistics
List of Sources
Nogier
WCAG1.0
WAI_G2
WAI_G1
"provide text alternatives for buttons defined with input tag"
"provide NOSCRIPT section when scripts are used"
"provide text alternatives for images via alt"
Jean

Results of WAI_G1 : "provide text alternatives for images via alt"

Ligne	Code html	Correction et commentaires
	Next	
14	<img src="../../Mes%20documents/Mes%20images/bank-appFlow.gif" alt="Bank Flow">	OK
	Next	
18	<img src="../../Mes%20documents/Mes%20images/11.jpg" alt="Image d'un verseau dans la nuit de décembre; Image d'un verseau dans la nuit de décembre ; Image d'un verseau dans la nuit de décembre ; Image d'un verseau dans la nuit de décembre ; Image d'un verseau dans la nuit de décembre ;Image d'un verseau dans la nuit de décembre">	WARNING : Image text equivalent is long, consider providing description via longdesc
	Next	
12	<img src="../../Mes%20documents/Mes%20images/013.gif">	ERROR : Image text equivalent is missing
	Next	
16	<img src="../../Mes%20documents/Mes%20images/bank-webFlow.gif" alt="">	WARNINGS : FINAL RESULT : SUCCESS STOP

Fig. 7. List of tested instances in a page

It is possible for the user to choose the format of the report as HTML is not imposed. For example, the evaluation report could also be formatted towards printing, such as in plain vanilla text format, in rich text format, and in PDF by automated translation into these formats. In this case, the user reduces the benefits of the navigation on the mini-site, but it is no longer intended to be used with the same level of flexibility as found in the mini-site. Other parameters can be selected such as the corresponding theory, whether to see the lines with the errors identified, etc.

5 Conclusion

In this paper we have presented an environment that supports both designers and evaluators in evaluating any set of quality properties of a web site (in particular, usability and accessibility, but not limited to). In addition, this environment enables the users to parameterize the usability report in such a way that is on-demand, dynamic, and flexible. A user survey has been conducted to determine the most frequently used format of such usability reports in usability organisations. The environment also introduces report navigation, compatibility with W3C standard EARL at level 1, and flexible visual presentation of the report. The profile of the user determines default values of the parameters used to write the usability report, but can be overwritten by custom values stored in a configuration file that can be saved for future usage. In addition, we are now testing the report usability with end users. We will analyse the results and take advantage of these to improve it. For the first time, it is possible to generate usability reports with an unprecedented level of flexibility such as: usability errors sorted by level of importance, of frequency, by page, by origin (e.g. accessibility vs. usability), attached to a web site, a series of web pages, a single page, a section of a page or even a page element. The level of details with which the usability error is reported is also flexible by incorporating more or less information coming from the guideline description, expressed in a XML-compliant language that serves for the computer-aided evaluation. Finally, any generated usability report can be sent through e-mail, viewed and navigated on-line.

Acknowledgement

We gratefully acknowledge the support of the DESTINE research/development project by the Walloon Region, under the WIST (Wallonie Information Société Technologies) convention n°315577.

References

1. Abascal, J., Arrue, M., Fajardo I., Garay, N., Tomas, J.: Use of Guidelines to Automatically Verify Web Accessibility. Universal Access in the Information Society 3,1 (2004) 71–79
2. ATRC: A-Prompt: Web Accessibility Verifier. Adaptive Technology Resource Center (University of Toronto) and Trace Center (University of Wisconsin), Canada & USA
3. Atterer, R.: Where Web Engineering Tool Support Ends: Building Usable Websites. In: Proc. of the 20th Annual ACM Symposium on Applied Computing SAC'2005 (Santa Fe, 13-17 March 2005). ACM Press, New York (2005)
4. Beirekdar, A., Vanderdonckt, J., Noirhomme-Fraiture, M.: A Framework and a Language for Usability Automatic Evaluation of Web Sites by Static Analysis of HTML Source Code. In: Proc. of 4th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2002 (Valenciennes, May 2002). Kluwer Academics Pub., Dordrecht (2002) 337–348
5. Brajnik, G.: Automatic Web Usability Evaluation: Where is the Limit? In: Proc. of the 6th Conf. on Human Factors & the Web (Austin, June 2000). Univ. of Texas, Austin (2000)

6. Brinck, T., Hermann, D., Minnebo, B., Hakim, A.: AccessEnable: A Tool for Evaluating Compliance with Accessibility Standards. In: CHI'2002 Workshop on Automatically Evaluating the Usability of Web Sites.
7. Ellis, R.D., Jankowski, T.B., Jasper, J.E., Tharuvai, B.S.: Listener: a Tool for Client-Side Investigation of Hypermedia Navigation Behavior. Behavior Research Methods, Instruments & Computers 30, 6 (1998) 573–582
8. Forrester Research: Why most web sites fail. 1999. Available at http://www.forrester.com/Research/ReportExcerpt/0,1082,1285,00.html
9. Macromedia. Macromedia Exchange - 508 Accessibility suite extension detail page (2001) ACM International Conference Proceedings Series, New York (2004) 117–124
10. Ivory, M.Y., Hearst, M.A.: State of the Art in Automating Usability Evaluation of User Interfaces. ACM Computing Surveys 33, 4 (2001) 470–516
11. Ivory, M.Y., Mankoff, J., Le, A.: Using Automated Tools to Improve Web Site Usage by Users with Diverse Abilities. IT&Society 1,3 (2003) 195–236
12. Ivory, M.Y. Automated Web Site Evaluation: Researcher's and Practitioner's Perspectives. Kluwer Academic Publishers, Dordrecht (2003).
13. Jackson-Sanborn, E., Odess-Harnish, K., Warren, N.: Website Accessibility: A Study of ADA Compliance. Technical Report TR-2001-05. School of Information and Library Science, University of North Carolina at Chapel Hill (2001)
14. Okada, H., Asahi, T.: An Automatic GUI Design Checking Tool from the Viewpoint of Design Standards. In: Proc. of 8th IFIP Conf. on Human-Computer Interaction Interact'2001. IOS Press (2001) 504–511
15. Paternò, F, Ballardin, G.: RemUSINE: a Bridge between Empirical and Model-based Evaluation when Evaluators and Users are Distant. Interacting with computers 13,2 (2000) 151–167
16. Pennsylvania's Initiative on Assistive Technology. Wave V 3.0. Available at http://www.wave.webaim.org/wave/index.jsp
17. Ridpath, C., Treviranus, J.: Integrated Accessibility Evaluation and Repair (The Development of A-Prompt). In: Proc. of CHI'2002 Workshop on Automatically Evaluating the Usability of Web Sites, CHI'2002, Minneapolis (2002)
18. Thatcher, J.: Section 508 Web Standards & WCAG Priority 1 Checkpoints: A side-by-side Comparison. Available at http://jimthatcher.com/sidebyside.htm (2002)
19. Vanderdonckt, J., Beirekdar, A., Noirhomme-Fraiture, M.: Automated Evaluation of Web Usability and Accessibility by Guideline Review. In: Proc. of 4th Int. Conf. on Web Engineering ICWE'04 (Munich, 28-30 July 2004), Springer-Verlag, Berlin (2004) 17–30
20. Vanderdonckt, J.: Development Milestones towards a Tool for Working with Guidelines. Interacting with Computers 12, 2 (December 1999) 81–118
21. W3C Web Content Accessibility Guidelines. Available at http://www.w3.org/