KWARESMI – Knowledge-based Web Automated Evaluation Tool with Reconfigurable Guidelines Optimization

Abdo Beirekdar[1,2], Jean Vanderdonckt[2], Monique Noirhomme-Fraiture[1]

[1]Facultés Universitaires Notre-Dame de la Paix, Institut d'Informatique
Rue Grandgagnage, 21 – B-5000 Namur (Belgium)
{abe,mno}@info.fundp.ac.be
[2]Université catholique de Louvain, BCHI
Place des Doyens, 1 – B-1348 Louvain-la-Neuve (Belgium)
{beirekdar,vanderdonckt}@isys.ucl.ac.be

Abstract

KWARESMI (Knowledge-based Web Automated and Reconfigurable Evaluation with guidelineS optiMIzation) tool is aimed for expressing, structuring, and organizing web usability guidelines towards automated evaluation. Traditional tools transform natural statements of usability guidelines into lines of code in a tool that parses the HTML code of a web page and performs guidelines review. These tools are inflexible by nature: impossible to introduce a new guideline, to modify an existing one, to select guidelines on demand before evaluation. Furthermore, there is no optimization of how guidelines can be evaluated in an efficient way: all guidelines are tested on all elements, without factoring out common parts. KWARESMI attempts to address these shortcomings by allowing evaluators to express guidelines in a higher level than simply HTML code and using them for automated usability evaluation on demand.

1 Introduction

Usability is today recognised as a major quality and success factor of web sites. A wide range of usability evaluation techniques have been proposed and many of them are currently in use (Ivory & Hearst, 2001). They range from formal usability testing to informal usability tests conducted by usability specialists at usability labs or among real users.

Automation of these techniques became much desired (Brajnik, 2000; Ivory & Hearst, 2001; Cooper, 1999) because they required usability specialists to conduct them or to analyse evaluation results, which is very resource consuming especially for very large, continuously growing web sites. In addition, there is a lack of usability and accessibility experts due to an increased demand. A possible solution consists of capturing the knowledge and experience of these experts and expressing it in form of recommendations or guidelines to be reviewed and applied by designers and developers. Some studies show that applying guidelines by designers is subject to interpretation, basically because of the inappropriate structuring or formulation (Scapin et al., 2000).

For this reason and others, automation has been predominately used to objectively check guideline conformance or review (Ivory & Hearst, 2001). Many automatic evaluation tools were developed to assist evaluators with guidelines review by automatically detecting and reporting ergonomic violation and making suggestions for repairing them. Representative examples of these tools include: A-Prompt (A-Prompt, 1999), LIFT (LIFT, 2003), Bobby (Cooper, 1999) and WebSat (Scholtz & Laskowski,, 1998). Some tools can be integrated with popular web design tools and

methods (LIFT, 2003). The most popular set of guidelines evaluated by most existing evaluation tools are the W3C Web content Accessibility Guidelines (http://www.w3c.org/TR/WC AG10) and Section508 guidelines (http://www. section508.gov).
In this paper, we present a tool that overcomes major shortcomings of existing evaluation tools. It enables evaluators to express the ergonomic body of knowledge provided by guidelines in terms of HTML elements (tags and attributes). Once coded, this knowledge can be evaluated dynamically at evaluation-time by configuring the guidelines expressions in an optimised way depending on the guidelines to be evaluated and the elements contained in the page. This process consequently considers guidelines 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.
This paper is structured as follows: section 2 gives detailed description of the tool. Section 3 concludes the paper by underlying major potential advantages of the proposed tool.

2 The KWARESMI Tool

Our tool tries to overcome major shortcomings of existing automatic evaluation tools due to hard coding the evaluation logic inside their evaluation engine.

2.1 Tool Requirements

The tool is supposed to meet the following requirements:

A. *Be knowledge-based*: it exploits the ergonomic knowledge contained in ergonomic guidelines and re-expresses it in terms of HTML knowledge contained in the semantic of HTML elements.
B. *Be Web-oriented*: it works on HTML code of Web pages. At first step we target Web pages containing HTML only (no CSS, Scripts, etc.).
C. *Be automated*: it should automate the maximum of tasks related to Web evaluation process, the main task being the evaluation itself.
D. *Be reconfigurable*: the tool must enable evaluators to control the manipulation of both the guidelines structuring and the evaluation process.
E. *Enable guidelines optimization*: the tool should identify potential common ergonomic information among different guidelines. This information should be used to optimize evaluation of these guidelines.
F. *Support the GDL formal language*: the development of KWARESMI was triggered by the need for a tool to support the Guideline Definition Language that we propose to structure Web guidelines towards automatic evaluation.

It is worth noting that KWARESMI is intended to enable the evaluation of any ergonomic guideline (for usability, for accessibility, etc.) as soon as we can find HTML elements that enable the evaluation of this guideline partially or totally. Therefore, the evaluation should not be restricted to specific types of guidelines.

2.2 Tool architecture

Figure 1 shows the software architecture of our tool. A data base is used to store all provided or generated information to ensure their reusability for future evaluation sessions. A detailed description of these concepts is given in (Beirekdar, Vanderdonckt & Noirhomme, 2002).

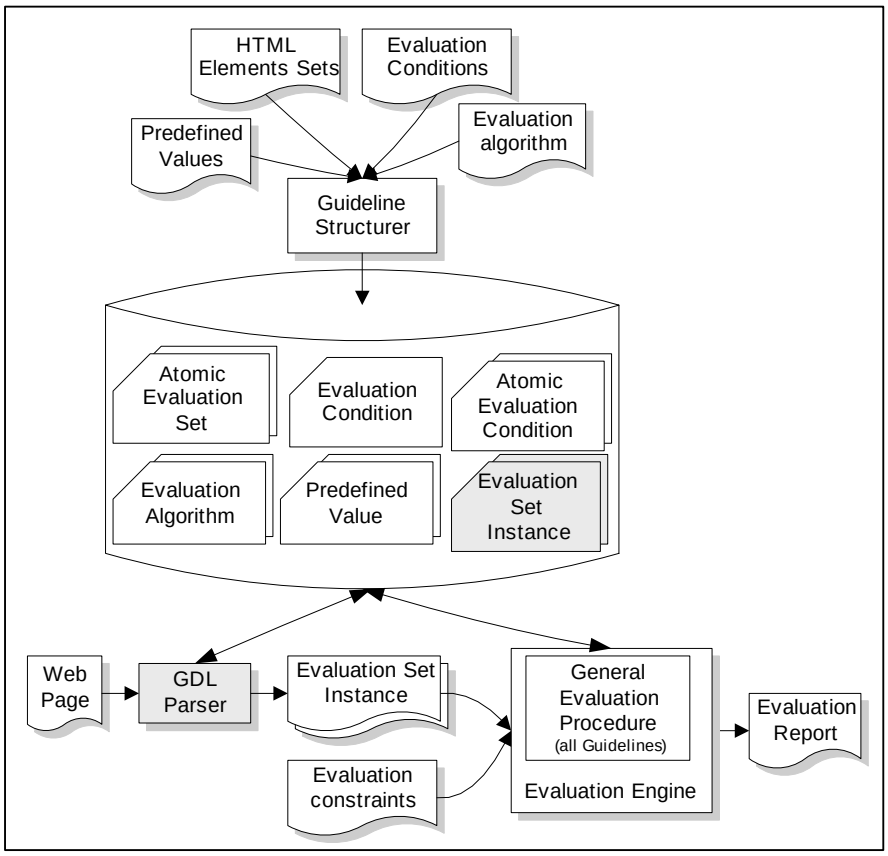

Figure 1: KWARESMI software architecture

The tool is composed of three main modules:

2.2.1 Guideline structuring module

Instead of hard coding the evaluation logic of guidelines in the evaluation engine, KWARESMI enables the evaluator to express guidelines in an evaluation-oriented manner. The new guidelines structure is stored in the tool data base, thus separated from the evaluation engine. The structuring is realized as follows:

- The Guideline Structurer is used to re-express guidelines in terms of HTML elements that are grouped into *evaluation sets*. Each set represents the zone of a Web page to be examined to evaluate the whole (or a part of the) treated guideline.
- These evaluation sets are then used to define *evaluation conditions* that determine when to consider that the evaluation set is respected or violated.
- Evaluation sets and evaluation conditions are then used to define the *evaluation algorithm* that describes how to evaluate the whole guideline. All this information is stored in the tool data base to be used or modified later, thus ensuring more flexibility in controlling the evaluation logic.

2.2.2 Web page parsing module

It is obvious that most of the time we do not need to parse the whole Web page to evaluate a given guideline. For example, to evaluate the guideline that recommends providing alternative text for images, all what we need is to check occurrences of IMG tag. Our parsing strategy is based on scanning the Web page and extracting only instances of the tags and attributes that we used when structuring our guidelines in term of evaluation sets.

2.2.3 Evaluation module

To evaluate the Web page, we use the evaluation algorithm that was defined to determine the order of checking the evaluation sets. Checking an evaluation set is done by checking its instances that were detected in the parsed page to see if they respect or not the evaluation condition that was associated with the evaluation set.

2.3 Tool implementation

KWARESMI prototype (Figure 2) is implemented using Java language to ensure future portability. We needed to define our HTML specific parser to ensure optimal and configurable parsing of the evaluated Web page. Currently, KWARESMI works locally as an off-line tool. Next evolution would be to make it accessible via a Web site as an on-line evaluation service.

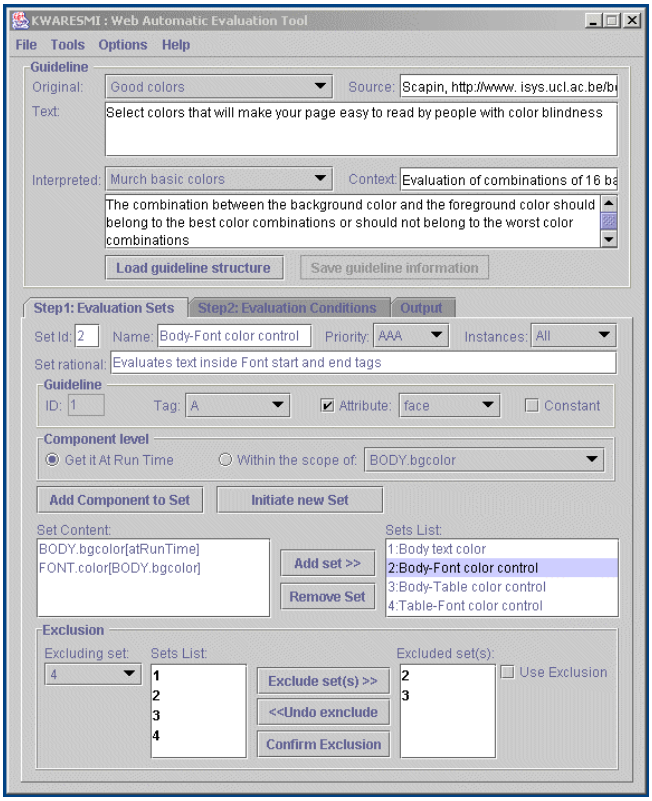

Figure 2: Structuring guidelines with the KWARESMI tool

3 Conclusion and perspectives

In this paper we presented our KWARESMI tool we are developing to support our GDL as an automatic evaluation tool of Web ergonomic guidelines. The tool and the underlying GDL would overcome main shortcomings of existing automatic evaluation tools:

- The tool should allow the dynamic evaluation of any guideline that can be expressed in GDL without any code modification of the evaluation engine.
- It should enable an evaluator to define and redefine evaluation logic of a guideline if needed, again, without the need to modify and recompile the evaluation engine.
- It should enable the evaluator to realize on-demand evaluation by providing him by high control level of the evaluation process.

When applied on simple examples, the developed prototype produced promising results. With the advantage of the tool and the GDL language, there is now a possibility of introducing, managing, structuring, and evaluating guidelines at a level of abstraction that is higher than the one of HTML level. Of course, there is a long trip to transform a guideline expressed in natural language to its equivalent code working on HTML tags. This is why GDL and KWARESMI are located between to bridge the gap between a high level of abstraction (natural language) and a low level of implementation (HTML).

It is worth to note that currently implemented prototype covers guidelines evaluation in a single page only. It can not be used for guidelines that cover inter-pages relations or link-structure issues. Extending the tool toward this direction is planned after validating results for one page evaluation.

References

A-Prompt (1999). A-Prompt Accessibility Toolkit Project. Adaptive Technology Resource Center (University of Toronto) and Trade Center (University of Wisconsin).

Beirekdar, A., Vanderdonckt, J. & Noirhomme-Fraiture, M. (2002). A Framework and a Language for Usability Automatic Evaluation of Web Sites by Static Analysis of HTML Source Code. In Kolski, Ch. & Vanderdonckt, J. (eds.): Proceedings of 4th International Conference on Computer-Aided Design of User Interfaces.

Brajnik, G. (2000). Automatic Web Usability Evaluation: Where is the Limit? Kortum, Ph., Kudzinger, E. (eds.): Proc. of 6th Conf. on Human Factors and the Web HFWeb.

Cooper, M. (1999). Evaluating Accessibility and Usability of Web Pages. In Proceedings of 3rd International Conference on Computer-Aided Design of User Interfaces.

Ivory, M.Y. & Hearst, M.A. (2001). State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Computing Surveys*, **33,**4.

LIFT (2003). Retrieved February 3 2003, from http://www. usablenet.com.

Scapin, D.L., Leulier, C., Vanderdonckt, J., Mariage, C., Bastien, Ch., Farenc, Ch. & Palanque, Ph., Bastide, R. (2000). A Framework for Organizing Web Usability Guidelines. Kortum, Ph., Kudzinger, E. (eds.). In Proceedings of 6th Conference on Human Factors and the Web.

Scholtz, J. & Laskowski, Sh. (1998). Developing Usability Tools and Techniques for Designing and TestingWeb Sites. In Proceedings of the 4th Conference on Human Factors and the Web.