Adding Usability to Web Engineering Models and Tools

Richard Atterer[1] and Albrecht Schmidt[2]

[1] Media Informatics Group
Ludwig-Maximilians-University Munich, Germany
richard.atterer@ifi.lmu.de
[2] Embedded Interaction Research Group
Ludwig-Maximilians-University Munich, Germany
albrecht.schmidt@acm.org

Abstract. In this paper, we examine how the task of creating usable websites can be made more efficient. Models and generation of websites have been a central issue for Web Engineering over recent years. However, usability tool integration has not been a primary focus – few usability validators take advantage of models which describe the website. After a look at existing tools, we examine how information stored in models can help to improve validation. Furthermore, we highlight additional properties which, if present in models, would improve validation quality. We present the prototype of a model-based usability validator. Given the presentation model of an existing web page, it verifies a set of guidelines. Web Engineering methods need to take usability into account at many levels. Beyond the extension of models, this requires further semi-automated and manual steps for user testing.

1 Introduction

A basic demand of any website is that its *web application must work* in the sense that it must be possible to use the website for its intended purpose – this is addressed by research into Web Engineering methods, models and tools. However, there is also the equally important demand that the *web application must be usable* by the visitors of the website. Usability research includes work on sets of guidelines which help to improve website usability. Additionally, established procedures like user tests provide information about issues which make a site difficult to use. In this paper, we build upon existing research results in both of the above areas. We have analysed state-of-the-art Web Engineering solutions in previous work [2] and concluded that usability has not been their primary focus so far. Regarding usability, we have looked at a variety of different sources, including the W3C's Web Accessibility Initiative (WAI) and related documents [9], the Yale Web Style Guide [8] and Jakob Nielsen's alertbox series [7].

So far, research effort has concentrated *either* on development of Web Engineering models (and associated page generation tools) *or* on usability/accessibility validators, despite the fact that the information stored in Web

Engineering models could be very useful to these validators. The main contribution of this work is the analysis of how usability validation of websites can be improved when abstract information from models is available.

Section 2 of this paper examines the limits inherent in validating HTML pages without having a model which describes certain properties of these pages. Section 3 highlights the benefits of model-based usability validation in a systematic approach, taking into account presentational, navigational and functional aspects, and section 4 lists a number of proposed model extensions. After the presentation of the prototype validator in section 5, section 6 discusses the benefits of combining knowledge from the fields of Web Engineering and usability research, and presents some areas which should be addressed in future research.

2 Current Usability Validation Approaches

There exists a large number of usability and accessibility guidelines which is validated by current tools just by analysing the HTML pages, CSS (cascading style sheets) and other content that can be retrieved from a website.

However, the implementation of checks for these guidelines often suffers from the problem that no model is available, i.e. no abstract description of certain properties of the web page (or its parts). This way, the validator either fails to find certain usability problems in the pages or it outputs too many general warning messages. For instance, it is straightforward to check given HTML code for high colour contrast [4] and the use of a limited number of different font faces, but it is not possible to do this reliably for images which contain a rendered version of some text, unless a model provides information regarding the text contained in the image.

As part of our research, we have looked at the following usability and accessibility validators:

- A-Prompt (`http://aprompt.snow.utoronto.ca`)
- Bobby (`http://bobby.watchfire.com`)
- EvalIris [1]
- Kwaresmi (`http://www.isys.ucl.ac.be/bchi/research/Kwaresmi.htm`)
- LIFT (`http://www.usablenet.com`)
- NAUTICUS (`http://giove.cnuce.cnr.it/nauticus/nauticus.html`)
- WAVE (`http://wave.webaim.org`)
- WebTango [5]

None of these tools works with a presentational or navigational model taken from a Web Engineering solution like UWE [6] or OO-H [3]. Furthermore, none allows interactive "reverse-engineering" of models from existing web pages, or annotating them with abstract information like the tool prototype we present in section 5. Looking at the output of the tools, it becomes clear that the lack of additional, more abstract information about the pages is a problem: Many tools output messages which tell the user to perform manual checks for some of the page content.

3 Automatic Usability Checking Based on Models

The quality of automated usability tool support can be increased significantly by taking advantage of the models which are available in current Web Engineering solutions. For instance, UWE and OO-H both feature navigational models which provide details on the ways the site is intended to be traversed, and presentational models which define abstract properties of the page layout – for example, they allow us to assign meaning to parts of the page layout, like "this is advertisement". Due to space constraints, this section only gives a few examples of possible improvements.

3.1 Presentational Aspects

Standard page layout With a model which describes the different page areas, we can check whether the page design follows one out of a number of de-facto standards, for example "three columns with header, site name at top, navigation at left, advertisement at right". Related to this, it is possible to check whether the layout of content is consistent across all pages.

Liquid layout Using the model, we can easily say which part of the page has the main content. Consequently, the rule that a page's width should adjust to the browser window width can be made more accurate: It is desirable that the *main content's width* increase with the browser window width.

Essential content Finally, a tool can alert the user if page areas with certain content are missing. Content which should normally be present on every page includes the page creator's identity, a "last changed" note and a link to the site's entry page. Additionally, a complex site's main page will benefit from the presence of a search facility, a "news"-style list of recently updated site content, and other similar items.

3.2 Navigational and Functional Aspects

Navigation paths A model-based tool can analyse the possible navigation paths of the site in a variety of ways. For instance, the click distance between arbitrary pages can be calculated. The web developer can subsequently specify e.g. "there should only be 3 clicks from the product view to the final 'thank you for buying' message".

Interaction patterns The models of current Web Engineering solutions feature support for certain patterns, such as a "guided tour", i.e. a series of pages connected with "previous" and "next" buttons. It is possible to offer tool support for automatic recognition of such patterns, e.g. by looking for sequential steps in the model's activity diagrams. This way, it is ensured that typical ways of interacting with a site use appropriate, established interaction patterns.

Intended audience The model for a website could specify properties of the site's intended audience. For instance, the audience can be assigned a "literacy" value, ranging from "children" to "academic person". An automated check can subsequently warn about site content which exceeds the target audience's vocabulary. Another example is the audience's type of Internet access – if it is slow, the pages must not contain too many large graphics.

4 Extending Web Engineering Models

If attributes related to usability are included in the Web Engineering models, this will allow tools to increase usability automatically or to warn the developer when certain guidelines are violated. We recommend that the following selection of attributes be included in Web Engineering models:

- Timing
  - Overall contact time of a user with the site?
  - Contact time per visit?
  - How long will the user need for the main tasks?
  - What is the maximum time for delivery of a page?
- Purpose of the site
  - What is the main objective of the web site?
  - What information and navigation complexity is desired?
  - Is the page mainly sensational, educational, or informational?
- Target group, anticipated user
  - What is the main user group?
  - Age distribution of the anticipated users.
  - Computer related skill level of potential users?
  - What infrastructure (e.g. computer type, connection speed) do potential users have?

Timing, site purpose and target group are central to many of the usability issues raised. The concrete attributes in these categories may vary depending on the models and Web Engineering system.

5 Prototype of a Model-Based Usability Validator

The implemented prototype of a model-based usability validator demonstrates some of the ideas presented in this paper. The most obvious difference from other validators is that the input to the program does not solely consist of a web page (or its URL), but that additional information about the page needs to be provided.

Like the majority of the other available validators, the tool is a server-side program with a web interface. The tool illustrates two concepts: The reverse-engineering of a (simplified) presentation model from a finished web page, and automated usability validation using that page model.

Fig. 1. Selection of a page area (highlighted) and classification as "main content" using the prototype usability validation tool.

After the user has supplied the URL of a web page, she or he is presented with a version of the page with some added controls (shown in figure 1). These controls allow the selection of regions of the page layout if it is based on HTML tables. Having selected a region, the user can assign a content type to it, for instance "main content", "navigation" or "advertising". After some or all parts of the page have been annotated, the tool can perform a number of tests and output a result screen which alerts the user to problems with the web page.

6 Discussion and Conclusion

How usability can benefit from Web Engineering It may be argued that automatic checks are not equivalent to what a suitably trained expert could achieve. Replacing the usability expert in the process is not the aim of our research. However, we believe that the addition of "usability support" to Web Engineering solutions would lead to improved usability for websites because otherwise, in practice *no* measures which improve usability would be employed.

Today, general interest by web developers in usability is not as high as it should be. However, if the tools used by a developer "get it right" by default and require specific actions to override these defaults, the quality of the results will be significantly enhanced – an effect which has already been observed with GUI editors.

With support built into the tools, the resulting web pages are consistent by default, and the automatic checks inherent in the process provide feedback about usability-related problems, with only little extra effort by the developer.

How Web Engineering can benefit from usability methods For the architects of Web Engineering solutions, the models are one of the most impor-

tant aspects in their work: On one hand, the more properties the models define for web pages and entire websites, the more powerful the respective tool support can be. On the other hand, more detailed models also result in more work for the web developer who needs to create them. It tends to be difficult to find the right balance between automatability and simple modelling.

With our work, we hope to provide a basis for further enhancements to Web Engineering tools and models. We show that additional information in the models will lead to better usability in the final website. This way, we hope that this paper will assist Web Engineering architects in deciding where extensions to their models make sense.

All in all, the bar for Web Engineering is raised not only to allow the creation of *functional* websites, but also the creation of *easy-to-use* ones.

Further work In our current and future work, we concentrate on tool support for improving usability in the context of Web Engineering, and investigate how a user centred design process can be combined with a Web Engineering process. This includes extensions to models and requires new steps in the development process. This paper is only one of the first steps in this direction; apart from implementing prototypes for model-supported usability validation, it must be evaluated in practice whether the proposed solutions result in better usability.

Acknowledgement This work was funded by the BMBF (intermedia project) and by the DFG (Embedded Interaction Research Group).

References

1. J. Abascal, M. Arrue, N. Garay, J. Tomás: EvalIris – A Web Service for Web Accessibility Evaluation. In *Proceedings of the 12th International World Wide Web Conference*, Budapest, Hungary, 20–24 May 2003.
2. R. Atterer: Where Web Engineering Tool Support Ends: Building Usable Websites. In *Proc. of the 20th Annual ACM Symposium on Applied Computing*, USA, 2005
3. C. Cachero, J. Gómez, O. Pastor: Object-Oriented Conceptual Modeling of Web Application Interfaces: the OO-HMethod Abstract Presentation Model. In *Proceedings of the 1st International Conference on Electronic Commerce and Web Technologies EC-Web 2000*, UK, pages 206–215, Springer LNCS 1875, 2000
4. F. H. Imai, N. Tsumura, Y. Miyake: Perceptual color difference metric for complex images based on Mahalanobis distance. Journal of Electronic Imaging – April 2001 – Volume 10, Issue 2, pages 385–393
5. M. Y. Ivory, R. R. Sinha, M. A. Hearst: Empirically Validated Web Page Design Metrics. In *Proceedings of the SIG-CHI on Human factors in computing systems*, March 31 – April 5, 2001, Seattle, WA, USA. ACM, 2001
6. A. Knapp, N. Koch, G. Zhang, H.-M. Hassler: Modeling Business Processes in Web Applications with ArgoUWE. In *Proceedings of the 7th International Conference on the Unified Modeling Language (UML2004)*. Springer Verlag, October 2004
7. J. Nielsen: Alertbox: Current Issues in Web Usability. `http://useit.com/alertbox/`, accessed 10 Feb 2004.
8. Web Style Guide `http://www.webstyleguide.com/`, accessed 5 November 2004.
9. World Wide Web Commitee (W3C): Web Accessibility Initiative (WAI), `http://www.w3.org/WAI/`, accessed 5 Feb 2005.