

# Merging Web Accessibility and Usability by Patterns

Helmut Vieritz, Daniel Schilberg, and Sabina Jeschke

Center for Learning and Knowledge Management/Department of Information  
Management in Mechanical Engineering, RWTH Aachen, 52068 Aachen, Germany  
`vieritz@zlw-ima.rwth-aachen.de`

**Abstract.** Nowadays, usability and accessibility are two important quality aspects of user interfaces (UIs). But in research and development, both are sometimes accounted as unity and sometimes as contradiction. Understanding Web accessibility as an essential part of usability, here an approach is given to clarify the relationship of both. Therefore,

- a pattern approach was chosen to meet the practical requirements of UI designers which are often not familiar with accessibility needs,
- a collection of patterns for human-computer interaction (HCI) has been extended with accessibility requirements and
- finally a pattern example illustrates the cooperation of usability and accessibility in detail.

## 1 Introduction

Usability and accessibility as well are two growing factors in product design. The use of modern technology is mostly not limited by product functionality but UI design and usage under limiting conditions. Limits can be set by users with special needs as people with disabilities or elderly people, technology as cellular phones etc. or an environment with noise or bright light.

But, the relationship of accessibility and usability is still not clarified. The usability definition most widely adopted was given by Nielsen [1] and is focused on learnability, efficiency, memorability, few errors and users' satisfaction. The common understanding of accessibility defines it as access to and controll of all information for all users. Even people with impairments can perceive, understand, navigate through and interact with the (Web) application [2]. Since these definitions are focused on different aspects of usage – general quality of usage versus access to information – the relationship is not simply given. Accessible applications have positive impacts for all users. Therefore, accessibility is often seen as a part of usability [3]. Integrating usability and accessibility issues in the software design and development process is a complex challenge. Additionally, different (Web) technologies raise the needed knowledge and costs.

The presented research has the objective to facilitate the integretation of accessibility requirements in practical work. Modern approaches of application development are focused on UI design and usability since the quality of UI has

a growing impact on the commercial success of products. But until now, accessibility is mostly not seen as an important commercial criterion for better product design and has not the same impact. A solution may be to see accessibility from usability point of view and to give more support for the early design process of UIs then existing recommendations as the *Web Content Accessibility Guidelines* (WCAG 2.0) [4] can do. The presented approach adapts the needs of accessibility to UI design which is focused on usability. Therefore, a pattern framework of interaction pattern has been extended with the requirements for accessible UIs.

## 2 Related Work

Patterns for usability and interaction design are broadly discussed in research and practice. Some frameworks already exist. Borchers [5] has developed and detailed discussed a pattern framework which is focused on an interactive music exhibition. Folmer et.al. [6] has published a framework of software architecture-relevant patterns and Graham [7] has published a pattern language for Web usability. Other pattern languages for interaction design can be found in [8,9,10]. In these frameworks, accessibility is sometimes mentioned but not integrated in detail. For example, in [6] accessibility is reduced to multimodal use. Few papers exist for Web accessibility patterns [11,12] which do not rely on usability. Abraham [13] presents a case-based reasoning (CBR) approach to improve the accessibility of interaction patterns. His approach does not clarify the relationship to accessibility standards as WCAG. In general, an overall approach to merge usability and accessibility is still missing.

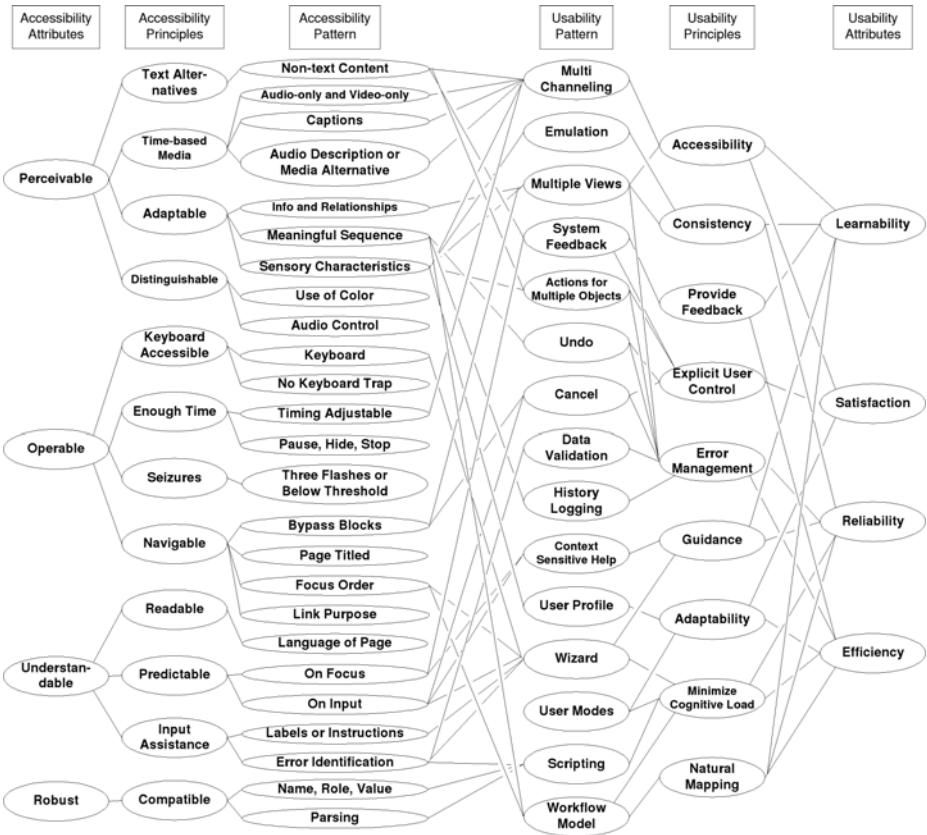
## 3 Accessibility and Usability by Patterns

Pattern languages are a well-suited method to explain basic idea without the technological overhead. Patterns have been introduced in urban architecture by Christopher Alexander in 1977 [14]. Nowadays, they are widely known in software development and HCI as well. An *usability* or *interaction pattern language* in HCI is a collection of recommendations for the design of usable UIs. A *pattern* is a preformal construct which gives a best practice solution for a repeating problem. It is described in a defined structure such as problem, context and solution. Typical categories are:

- a name for the pattern which is memorizable and unique
- a short statement of the problem
- a description of pattern context
- a solution for the pattern which explains the problem and its solution in detail
- a graphical depiction for pattern application which underlines the pattern idea

Additionally, a *pattern framework* includes structure and relationships between patterns. Until now, attempts of stronger formalization with e.g. the *Pattern Language Markup Language* (PLML) were not successful.

Folmer et. al. [6] give an overview for common usability patterns (see fig. 1). This framework was chosen for current research of accessibility influence on software architecture (not discussed here). *Usability attributes* are precise and measurable criteria of the abstract concept. There are four common attributes: efficiency, reliability, satisfaction and learnability. Heuristics and design principles are embodied in *usability properties*. Fifteen typical architecture-relevant usability patterns are described.



**Fig. 1.** A pattern collection for accessibility and usability (derived from [4,6], mappings reduced)

Using the WCAG 2.0-recommendations [4] a similar classification for accessibility patterns can be derived. The WCAG-guidelines are well-suited to form a pattern language since they are technology-independent and based on a long discussion between all interest groups. The four basic attributes are perceivable,

operable, understandable and robust. The twelve different main guidelines are taken as basic principles of accessibility. Here, for better overview only the 25 level A<sup>1</sup> guidelines are shown. Changing by the additional guidelines for level AA and AAA is small since they only enlarge often practical and technological efforts e.g. captions for live stream additional to pre-recorded multimedia content.

After selection and classification of usability and accessibility patterns they are mapped on each other. The mapping of an accessibility pattern to an interaction pattern is sometimes caused by the same idea of solution and sometimes by the same type of pattern context. Some accessibility patterns have a context which is valid in many situations. For better overview they are included without mapping showing that they have to be integrated by all usability patterns. The relationship between both is best presented from usability point of view. The mappings show which accessibility requirements have to be respected in the context of usability patterns. The absence of contradictions underlines that accessible control of user interfaces is not counterpart of usability. The usability pattern structure is extended with categories *Relevant Accessibility Patterns*, *Main Accessibility Issues* and an additional illustration to include the requirements of accessibility in UI design. These three additional categories explain and illustrate how the proposed solution for an usable UI design can be made accessible as well.

## 4 A Pattern Example

The usability pattern *Wizard* is given as an illustrating example. Wizards in UI design are well-known in many situations such as shipping and payment dialogs in online shops or dialogs for the conversion of file formats. A Wizard-situation is characterized by a high level of interactivity since the user needs some information and has to make decisions. Therefore, it include access to information and UI control as well. At first, the pattern example describes the organization of a wizard and completes the description with the accessibility requirements. More pattern from the library can be found on the project Web site <http://www.inamosys.de>. Figure 2 shows the accessibility patterns which are relevant for the Wizard pattern.

### 4.1 Pattern *Wizard*

#### Problem

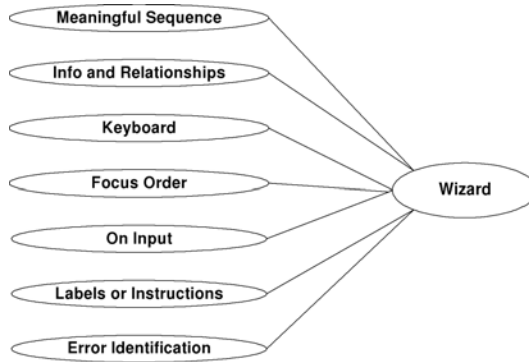
The user has to make several decisions to achieve a single goal. He may need additional information, previews or examples about the result of decisions.

#### Context

Infrequent complex tasks need to be performed by non-expert users. The task consists of several subtasks (3-10) each containing a decision. The user wants to

---

<sup>1</sup> WCAG have three conformance levels from A (lowest) to AAA (highest) to correspond to different needs of groups and situations.



**Fig. 2.** Accessibility patterns for interaction pattern *Wizard* (selection from fig. 1)

complete the task but may be not familiar with or interested in the subtasks. The single steps may rely on each other and they are ordered.

**Illustration**

amazon.com SIGN IN SHIPPING & PAYMENT GIFT-WRAP PLACE ORDER

Enter a new shipping address.  
When finished, click the "Continue" button.

Full Name: Ernest Hemingway

Address Line1: 907 Whitehead Street  
Street address, P.O. box, company name, c/o

Address Line2:   
Apartment, suite, unit, building, floor, etc.

City: Key West

State/Province/Region: Florida

ZIP/Postal Code: 33040

Country: United States

Phone Number:

Is this address also your billing address (the address that appears on your credit card or bank statement)? ☒ Yes ☐ No (If not, we'll ask you for it in a moment.)

Continue

**Fig. 3.** Wizard for shipping and payment on Amazons Web page

**Solution**

The user is taken through the entire task step by step (see fig. 3). Existing and completed steps are shown. The user is informed about the task goal from beginning on and that several decisions are to be made. Widgets are provided for

navigation through the entire task. The user can recognize that a subtask has to be fulfilled before the next is available. Revising of decisions is possible by back navigation to a previous subtask. The user is given feedback for the purpose of decisions. Explanations, previews or examples clarify the results. The final result is shown. If possible, appropriate default settings and short cuts accelerate and simplify the process for experienced users.

### Relevant Accessibility Pattern

Typically, a wizard uses forms to get needed information from the user. Buttons serve to finish the single steps. To be accessible, the different input fields of the form must be labeled, accessible by keyboard and the order of access must correspond with the workflow. Error identification and input help should be available and accessible as well.

Relevant accessibility patterns are (a) *Meaningful Sequence*, (b) *Keyboard*, (c) *Focus Order*, (d) *On Input*, (e) *Error Identification* and (f) *Labels or Instructions* (see fig. 4).

### Solution Illustration

The image shows a screenshot of the Amazon.com shipping address form. The form is titled "Enter a new shipping address." and includes a sub-instruction: "When finished, click the 'Continue' button." The form fields are as follows:

- Full Name:** Ernest Hemingway
- Address Line1:** 90 Whitehead Street
- Address Line2:** (empty)
- City:** New West
- State/Province/Region:** Florida
- ZIP/Postal Code:** 33440
- Country:** United States
- Phone Number:** (empty)

Below the form fields is a question: "Is this address also your billing address (the address that appears on your credit card or bank statement)?" with "Yes" and "No" radio buttons. The "No" option is selected, with a note: "(If not, we'll ask you for it in a moment.)". A "Continue" button is at the bottom right.

Annotations in blue boxes with arrows point to various parts of the form:

- (a) Meaningful Sequence:** Points to the "Enter a new shipping address." title.
- (f) Labels:** Points to the "Full Name:" label.
- (b) Keyboard:** Points to the "Continue" button.
- (c) Focus Order:** Points to the "City:" label.
- (d) On Input:** Points to the "Full Name:" input field.
- (e) Error Identification:** Points to the "Full Name:" input field.

Fig. 4. Accessibility requirements for wizard

### Main Accessibility Issues

Widgets are labeled and order corresponds with workflow. All functionality is available by keyboard. Name and role of interface components can be programmatically determined when scripts are used. User is advised in advance when changing the setting or focusing of user interface components does automatically cause a change of context.

## 5 Conclusions

The objective of the presented approach is the extension of an usability-driven UI design with the requirements of accessibility. A pattern approach was used to describe the main ideas independent from used UI platform. It gives software architects and developers an easy and quick way to understand the basic ideas of accessible user interfaces and helps them to see usability and accessibility not as a contradiction. The extension with new interaction patterns is easy.

The approach is still limited to be only a collection of pattern. Further work will be done on structure, completeness and relationships between pattern to built up a pattern framework which merges usability and accessibility in practical UI design. Furthermore, the pattern collection serves for analysis of accessibility influence on software architecture in usage-driven UI design.

## Acknowledgments

We highly appreciate Deutsche Forschungsgemeinschaft (DFG) for funding the INAMOSYS project.

## References

1. Nielsen, J.: Usability Engineering. Academic Press, Cambridge (1993)
2. Web Accessibility Initiative: Introduction to Web Accessibility (2005), <http://www.w3.org/WAI/intro/accessibility.php> (last visited: 03/11/2010)
3. Shneiderman, B.: Universal Usability. Communications of the ACM 43(5), 84–91 (2000)
4. World Wide Web Consortium: Web Content Accessibility Guidelines 2.0. (2008), <http://www.w3.org/TR/WCAG20/> (last visited: 03/11/2010)
5. Borchers, J.: A Pattern Approach to Interaction Design. John Wiley & Sons, Chichester (2001)
6. Folmer, E., van Gurp, J., Bosch, J.: Software Architecture Analysis of Usability. In: Bastide, R., Palanque, P., Roth, J. (eds.) DSV-IS 2004 and EHCI 2004. LNCS, vol. 3425, pp. 38–58. Springer, Heidelberg (2005)
7. Graham, I.: A Pattern Language for Web Usability. Addison-Wesley, Amsterdam (2003)
8. van Duyne, D.K., et al.: The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. Addison-Wesley, Amsterdam (2002)
9. Tidwell, J.: Designing Interfaces. O'Reilly Media, Sebastopol (2005)
10. Miller, F., Vandome, A., McBrewster, J.: Interaction Design. Alphascript Publishing (2009)
11. Ihmig, S.: Web-Accessibility Patterns. Master's thesis, Hamburg University (2007)
12. Vieritz, H., Jeschke, S., Pfeiffer, O.: Using Web Accessibility Patterns for Web Application Development. In: Proceedings of the 2009 ACM symposium on Applied Computing, pp. 129–135 (2009)
13. Abraham, G.: Needles in a Haystack. Improving Accessibility of Interaction Patterns (2006), <http://idea.library.drexel.edu/bitstream/1860/1569/1/2007021001.pdf>
14. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language. In: Towns, Buildings, Construction. Oxford University Press, New York (1977)