Transforming our World through Universal Design for Human Development
I. Garofolo et al. (Eds.)

doi:10.3233/SHTI220862

A Pattern Language for Inclusive Design: A Set of Patterns for Designing Reusable Accessible Solutions

Stefano VALTOLINA[a,1] and Alessandro Vivian SISTO[b]
[a] *Department of Computer Science, Università degli Studi di Miano, Italy*
[b] *Department of Computer Science, Università degli Studi di Miano, Italy*

Abstract. A common cause for inconsistent accessibility in the design of ICT applications is that designers work to develop optimal and satisfactory interfaces but they do not take care to comply with all users' needs and possibilities to access. The study presented in this paper stems from a lack of published evidence able to support designs in developing accessible interfaces for websites, mobile apps or conversational Interfaces (chatbot or voicebot). This work aims at investigating accessible design patterns that can be used by designers to implement applications accessible by multiple devices. To solve this problem, we propose a design pattern language able to provide WCAG compliant design out-of-the-box solutions. The navigation through the design patterns allows designers to gather existing solutions based on solid evidence and examples to develop inclusive applications accessible by web, mobile devices or conversational agents.

Keywords. Accessibility pattern language, HCI design patterns, Inclusive design, Web and Mobile design, Chatbot.

1. Introduction

Accessibility has always been a problem that designers tend to avoid dealing with it. Mainly because it is still unclear what makes up accessible interfaces and what developers need to know to get there.

From a legislative point of view, the Web Accessibility Initiative published the Web Content Accessibility Guideline (WCAG 2.1) [1] intending to provide a single shared standard for web content accessibility. Nevertheless, without clear practical suggestions or examples about what designers need to do for designing an accessible user interface (UI), the development of WCAG compliant applications can be challenging and expensive.

This paper aims at investigating a solution devised by exploiting design patterns strategies proposed in the HCI (Human-Machine Interaction) field [2-5]. The main research question we want to answer concerns how the design patterns can provide designers with techniques and strategies they need to be aware of when building accessible, inclusive interfaces for different devices and interactions styles.

[1] Corresponding Author: Stefano Valtolina, Department of Computer Science, Univesità degli Studi di Milano, via Celoria 18 20133 Milano – Italy, stefano.valtolina @unimi.it

An interaction design pattern is a general and re-applicable solution for a usability problem that occurs with recurrence in the design of user interfaces or the design of interaction strategies. To date, design patterns specially developed for suggesting WCAG compliant solutions do not exist in particular for the design of multi-device devices such as websites, mobile applications and conversational agents (also known as chatbots voicebots). A conversational agent is a dialogue system that responds automatically using human language usually employing a chat-based or voice-based bot over the internet or as a portable device assistant.

For this reason, after Section 2 described accessibility issues that generally affect the design of inclusive applications, we propose in Sections 3 and 4 a pattern language for helping designers in dealing with accessibility design problems. Section 3 discusses the main works and theories in the fields of HCI design patterns. Section 4 focuses on presenting a design pattern language as an organized and coherent set of patterns, each of which describes a problem and the core of a solution. Problems are described through high-level design patterns that are solved by evaluating low-level design patterns.

To define our design pattern language, we extended the structure proposed by Jan O. Borchers in [4, 5]. This extension aims to enrich the pattern schema with a section dedicated to incorrect examples for illustrating recurring error situations; a section used to show which points of the WCAG are satisfied with the pattern; three sections that describe technical solutions specifically designed for websites, mobile applications and conversational interfaces. Section 4 also presents a practical example of the application of our design pattern language to develop an application for registering user data through a form. In section 5 we draw some conclusions and discuss the main problems we dealt with in the discovery of multi-device accessibility solutions.

2. Accessibility issues to design inclusive applications

One billion people, or 15% of the world's population, experience some form of disability, and disability prevalence is higher in developing countries[2]. The number of web accessibility lawsuits continues to steadily rise in early 2021[3]. General trends show an increase in lawsuits filed in federal courts and state courts. These two pieces of data alone should convince us of the importance of designing for accessibility.

Accessible features can be used not only to ensure access to people with disabilities but also to make webpages more usable both by people using older computers and by people using the latest technologies (such as personal digital assistants, handheld computers, or web-enabled cellular phones). This rule is also regulated at the legislative level. Regulations such as the Americans with Disability Act (ADA) [6] or the EU Web Accessibility Directive [7], say how public websites or applications have to be accessible for visitors of all abilities. By analyzing these regulations, we can see that they stem or are an extension of the WCAG (Web Content Accessibility Guidelines) [1], which is an international standard recognized as the benchmark for web accessibility.

Creating a truly accessible website requires expert assistance and testing. For this reason, many businesses assume that the investment will not yield a return immediately. This is one of the most common misconceptions. What we want to demonstrate in this paper is that it is not true that web accessibility design necessarily implies an expensive

[2] https://www.worldbank.org/en/topic/disability#1

[3] https://www.essentialaccessibility.com/blog/web-accessibility-lawsuits

or unnecessary addition to a development project. Nevertheless, web accessibility has indeed associated expenses in particular when you have to retrofit a website for adding accessible features. Despite this extra expenditure, correcting an already inaccessible site is always beneficial in the end since accessible sites are easier and cheaper to maintain. On the other side, to make from scratch a website that is accessible could cost virtually the same as developing one that is not.

According to these considerations, in this paper, we propose a design strategy based on the use of HCI designs patterns specifically designed to support designers in the development of accessible websites, mobile applications and conversational agents.

3. HCI design patterns

The concept of design pattern has been proposed as a solution to help designers in developing more usable systems [8,9,10]. Design patterns were created in the field of urban planning and architecture and subsequently became popular within the software engineering community [11].

The basic idea is to acquire information about frequent problems and how to solve them. Patterns aim to capture and communicate the best practices of user interface design with a focus on the user's experience and the context of use. As a result, they are an attractive technique, with interesting ramifications for designing across a variety of contexts.

Design patterns made their appearance in the software engineering community when Gamma et al. published one of the best-selling books on software engineering [12]. Riding on the success of this work, many design patterns and design pattern languages have been published in the field of software engineering. In recent years, HCI design patterns have become increasingly important and are frequently used for the design of successful user interfaces and user experiences [2-5]. An HCI design pattern describes a recurring problem together with a proven solution in the field of Human-Computer Interaction discipline. An HCI design pattern has a well-defined form, which is dependent on the individual author's preferences. Besides the publication of books, it is possible to find several works on the World Wide Web to disseminate and publish HCI design patterns across the HCI community.

On the web, many repositories are published including Jennifer Tidwell's pattern language for HCI design [2], UI Patterns – User Interface Design Pattern Library [13], Yahoo!Design Pattern Library [14], Welie.com – Patterns in Interaction Design [15]. Together with these design pattern repositories, several portals exist that provide collections of references to design pattern resources such as The Interaction Design Patterns Page [16], hcipatterns.org [17], The Pattern Gallery [18].

A pattern form follows the specs defined in its pattern language or pattern collection. Regardless of its form, a pattern is designed for facilitating the users in understanding the problem, context, and solution of a specific problem. The pattern itself, when it is part of a collection or a pattern language, may have references to other patterns. These relationships are used to provide users with a strategy to navigate through interconnected patterns that solve different parts of the same problem. Borchers in [3,5], proposes to organize the pattern language by using a graph structure to visually represent their interdependencies. This hierarchical structure leads the designer from patterns that solve large-scale design problems, to patterns that concern small design details and helps her/him to quickly identify related patterns.

In this paper, we aim at proposing a language of HCI design patterns that goes beyond Brocher's definition to create multi-device accessible applications. The language can be considered a universal design resource, to help designers in creating accessible graphical interfaces for web, mobile and chatbot/voicebot applications that comply with WCAG guidelines. The representation of knowledge through design patterns aims to reflect the problem-solving approach usually followed by designers, while the hierarchical organization of patterns defines a structured language for guiding designers throughout the design process.

4. HCI design patterns for designing multi-device accessible applications

This section illustrates the design pattern we defined by levering Borchers' language. Our extension aims at enriching the language structure with different sections. The first one focuses on incorrect examples to illustrate recurring error situations, the second describes which WCAG points are satisfied by the pattern and the third section is dedicated to presenting technical realizations of possible multi-device solutions. The last section is dived into three parts that are respectively dedicated to describing solutions for the websites, mobile applications and chatbots. Therefore, each pattern has the following components:

- **Name**: It describes the intention of the pattern, facilitating the navigation in the language and making the pattern distinguishable.
- **WCAG 2.1 checkpoints**: List of WCAG points that are met thanks to the application of the pattern.
- **Problem**: A brief statement of the problem addressed.
- **Sensitizing image**: An image that introduces the topic and the solution proposed by the pattern.
- **When to use**: It defines the situations compatible with the application of the pattern.
- **Solution**: It briefly illustrates the solution to the problem presented.
- **How to use**: It describes in detail each point that makes up the pattern solution.
- **Why to use**: It provides the reasons that motivate the application of the pattern.
- **Examples**: Practical examples of the application of the techniques that make up the solution.
- **What not to do**: Examples of error situations that lead to solutions that do not comply with the WCAG directives that the pattern must comply with.
- **Patter application - Websites**: Techniques dedicated to using the pattern if the graphical interface in question is part of a website.
- **Patter application - Mobile devices**: Techniques dedicated to using the pattern in case the graphic interface in question is part of a mobile application.
- **Patter application - Chatbot/Voicebot**: Techniques dedicated to using the pattern if the graphical interface in question is part of a chatbot or voicebot.

Based on the structure proposed in [8] and according to the type of problem dealt with, we have defined a hierarchical structure by breaking down a tree into three sub-trees defined as follows and depicted in Figure 1:

- **Multi-format presentation**: It deals with patterns related to the presentation and organization of contents of different formats.

- **Navigation**: It defines patterns useful to provide a correct and simple navigation interface.
- **Interaction**: Set of patterns that deal with the various problems that a user might face when interacting with a graphical interface.

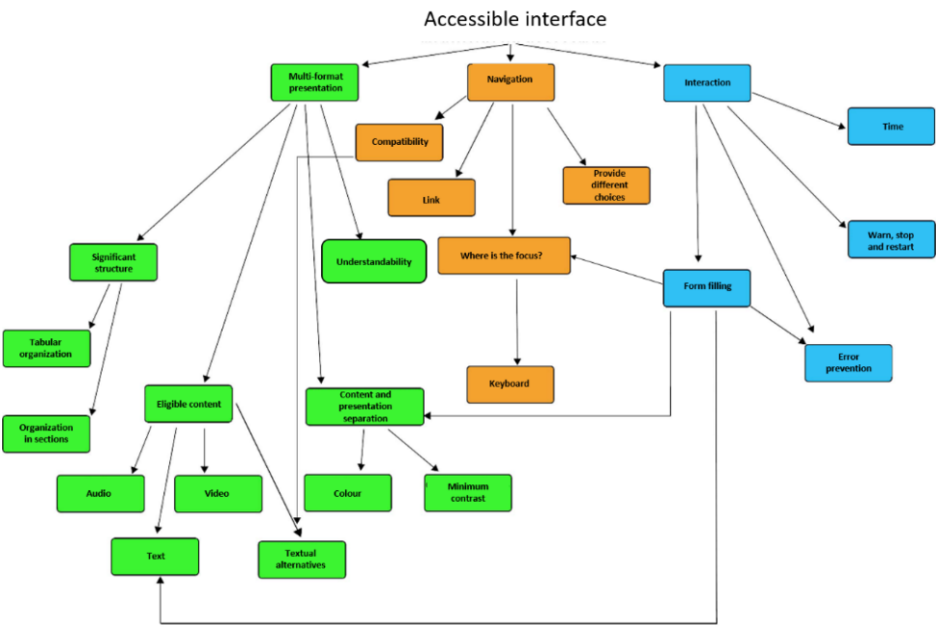

Figure 1. The structure of the tree. It represents the HCI pattern language.

4.1. How to navigate the design pattern tree

The navigation in the design pattern language starts from the root of one of the three sub-trees according to the graphic interface to create a multi-format presentation, navigation or interaction. In practice, once the designer identifies a leaf pattern related to the specific problem to solve, she/he has to go up the hierarchy to the root of the sub-tree. From the identified root pattern, the designer has to follow the hierarchy of the tree recursively applying the child patterns. Thanks to this categorization, the designer can focus on one aspect of the interface at a time, which is then preparatory to the application of the following category.

To connect the patterns, we defined relations of parent-child type as specified in [8]. This type of relationship defines a dependency between parent and child, in the sense that the correct application of the parent pattern depends on the correct application of its child pattern. However, different types of relationships are also possible, for example between the "*Compatibility"* pattern and the "*Textual alternatives"* pattern in Figure 1, there is a relationship that, while being of type parent-child, connects patterns of different sub-trees. These types of relationships allow designers to follow different strategies of design interconnecting navigation or interaction issues or problems related to the

presentation of the content for reusing the proposed solutions across many applications without reinventing the wheel.

4.2. Chatbot design accessibility: the main problem

With a growing number of businesses now using chatbots, it is now clear how the design of accessible chatbots is more important than ever. If the designer's objective is to develop a chatbot, the temptation of using a completely pre-built chatbot platform is very high. Nevertheless, this solution may not provide out-of-the-box accessibility support. Chatbots can present a range of difficulties for users. All of the graphical and textual content needs to fit together in a way that creates obvious connections and makes it intuitive to navigate to and within the chat. Screen readers need to be able to notify users of the context of chatbot conversations and replies. The chatbots are often programmed to speak, play music, use pop-up windows, and require users to click images to activate them. However, all content must be perceivable.

In our work, we investigated several online resources to detect accessible solutions for implementing different parts of a chatbot. These parts compose the final solution and can be explored by navigating the patterns presented in our design pattern language.

For example, regarding the "*Navigation*" pattern, if the user cannot use the mouse, the designer has to check if it is possible to click on all the buttons ("*Tab key*" to move the focus, "*Enter key*" to activate a button) for typing and sending messages. Moreover, it is necessary to move the scrollbar with the arrow keys (see Figure 2). To do it, the designer has to make this element focusable using a "*tabindex*" attribute.

As described in this example, although solutions to make a chatbot accessible exist it is not simple to find all possible indications to meet the WCAG criteria. In our proposal, we discovered solutions for only a part of the patterns composing our language. At the moment, we are working to complete the design pattern language but it is an expensive activity that needs more time to deal with.

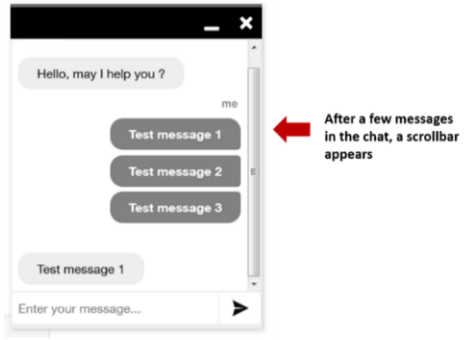

Figure 2. Chatbot screenshot that presents the appearance of a scrollbar. The designer has to find a solution for allowing the user to use the arrow keys for scrolling the messages.

4.3. Example of use

This paragraph shows an example of how to use the HCI patterns for accessibility to design a registration form for websites, mobile applications, and chatbots.

First, the designer must identify the "*Form filling*" pattern in the language tree (see Figure 1). Going up the trees, the reference root is the "*Interaction*" pattern. To develop an accessible application form it is necessary to recursively implement all patterns connected to the identified pattern up to the leaves of the design language tree. The pattern "*Form filling*" depicted in Figure 3 presents the main sections that characterize the pattern. For implementing a complete solution, the designer has to explore the design language tree along with the relationships that connect the pattern "*Form Filling*" to the other patterns: "*Content and presentation separation", "Colour", "Minimum contrast", "Text", "Where is the focus?"* and *"Error prevention"*. For each pattern, the designer can find solutions to develop a website or a mobile application or a conversational agent according to her/his needs. The patterns present what to do and what to not do for designing a final application that could be compliant with the accessibility criteria.

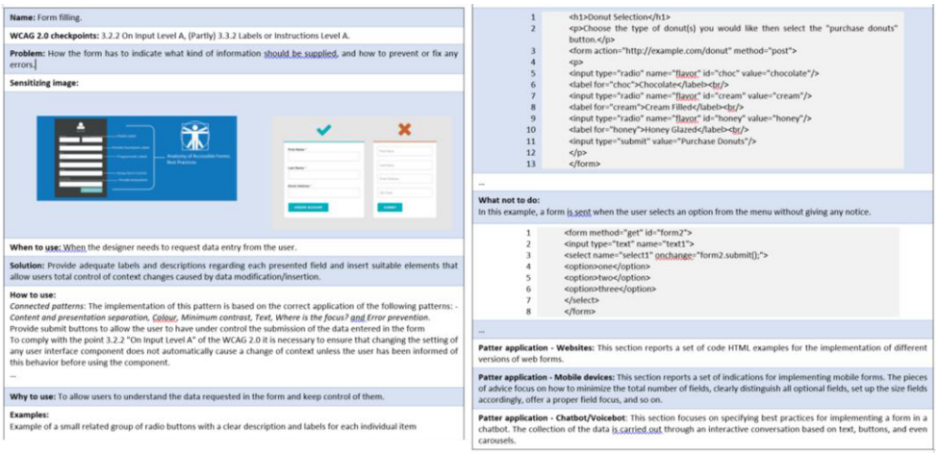

Figure 3. An example of a design pattern: *Form filling*. For space reasons, the schema reports only some indications of the total information that the final pattern has to contain.

5. Conclusion and discussion

This paper aims to present a language of HCI design patterns oriented toward multi-device accessibility. The idea is to allow people both with disabilities and not to take full advantage of the contents provided by websites, mobile applications, and conversational agents such as chatbots or voicebots.

In trying to eliminate the Web Accessibility Divide, or the gap between those who can independently access web resources and those who cannot, this language aims to offer a tool to simplify the work of designers by avoiding applying accessibility criteria after the development of the software product.

The main problem in creating accessible and multi-device patterns is the identification of solutions that can comply with the accessibility requirements of the WCAG. For what concerns the website's design strategies, the checkpoints described in the WCAG 2.1 specifications are sufficient for identifying solutions to be detailed in the patterns.

Instead, for the design of mobile applications and chatbot/voicebot the solutions must be identified using online resources not always created to satisfy accessibility requirements. This involves a long and expensive investigation that affects the definition

of patterns also in consideration of the rapid change in technological solutions that characterizes the design of mobile applications and conversational assistants.

Furthermore, the first draft of WCAG 3.0 has recently been published [19], which aims to make the language as simple as possible, so that even people who are not technology experts can understand it. From the first reading of this draft, WCAG 3.0 will offer many indications for making the web and other digital content (such as videos or mobile apps) more accessible to people. However, there is also no mention of solutions specifically designed for conversational agents.

References

[1] A. Kirkpatrick, J. O. Connor, A. Campbell, and M. Cooper. (Jul. 2018). Web Content Accessibility Guidelines (WCAG) 2.1. World Wide Web Consortium. [Online]. Available: https://www.w3.org/TR/WCAG21/ - last access 2022-04.
[2] Tidwell, J.: Designing interfaces—patterns for effective interaction design, 2nd edn. O'Reilly Media Inc., Sebastopol (2011).
[3] Kruschitz, C., & Hitz, M. (2010). Human-computer interaction design patterns: structure, methods, and tools. Int. J. Adv. Softw, 3(1).
[4] Borchers, J. O. (2008). A pattern approach to interaction design. In Cognition, Communication and Interaction (pp. 114-131). Springer, London.
[5] Borchers, J. O. (2000, April). Interaction design patterns: twelve theses. In Workshop, The Hague (Vol. 2, p. 3).
[6] https://adata.org/learn-about-ada - last access 2022-04.
[7] https://digital-strategy.ec.europa.eu/en/policies/web-accessibility - last access 2022-04.
[8] Fogli, D., Provenza, L. P., & Bernareggi, C. (2014). A universal design resource for rich Internet applications based on design patterns. Universal access in the information society, 13(2), 205-226.
[9] Bernhaupt, R., Winckler, M., Pontico, F.: Are user interface pattern languages usable? A report from the Trenches. In: Gross, T., et al. (eds.) INTERACT 2009, Part II, LNCS 5727, pp. 542–545. Springer, Berlin (2009).
[10] Cowley, N.L.O., Wesson, J.L.: An experiment to measure the usefulness of patterns in the interaction design process. In: Costabile, M.F., Paternò, F. (eds.) INTERACT 2005, LNCS 3585, pp. 1142–1145. Springer, Berlin (2005).
[11] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl- King, I., Angel, S.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press, UK (1977).
[12] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Re-usable Object-Oriented Software. Addison- Wesley, Reading (1995).
[13] "UI Patterns - User Interface Design Pattern Library", Available at: http://ui-patterns.com, last access 2022-04.
[14] "Yahoo! Design Pattern Library", Available at: http://developer.yahoo.com/ypatterns/, last access 2022-04.
[15] "Welie.com – Patterns in Interaction Design", Available at: http://www.welie.com, last access 2022-04
[16] Erickson, T., "The Interaction Design Patterns Page", Available at: http://www.visi.com/~snowfall/InteractionPatterns.html, last access 2022-04.
[17] "HCIPATTERNS.ORG", Available at: http://www.hcipatterns.org/, last access 2022-04.
[18] Fincher, S., "The Pattern Gallery", Available at: http://www.cs.kent.ac.uk/people/staff/saf/patterns/gallery.html, last access 2022-04.
[19] HCI patterns. URL: https://hcipatterns.org/patterns.html, last access 2022-04.