Adaptive User Interfaces for People with Cognitive Disabilities within the Easy Reading Framework

Peter Heumader^(✉), Klaus Miesenberger,
and Tomas Murillo-Morales

Institut Integriert Studieren, Johannes Kepler Universität Linz,
Altenbergerstraße 69, 4040 Linz, Austria
{peter.heumader,klaus.miesenberger,
tomas.murillo_morales}@jku.at

Abstract. Adaptive user interfaces are user interfaces that dynamically adapt to the users' preferences and abilities. These user interfaces have great potential to improve accessibility of user interfaces for people with cognitive disabilities. However automatic changes to user interfaces driven by adaptivity are also in contradiction to accessibility guidelines, as consistence of user interfaces is of utmost importance for people with cognitive disabilities. This paper describes how such user interfaces are implemented within the Easy Reading framework, a framework to improve the accessibility of web-pages for people with cognitive disabilities.

Keywords: Cognitive accessibility · Adaptive user interfaces · Web-accessibility

1 Introduction

The concept of user interfaces that have the ability to change according to the user's requirements, skills, environment, situation, or other criteria has been around for a long time. In general, these concepts can be categorized in adaptive user interfaces and adaptable user interfaces.

- Adaptive User Interfaces [1]: These systems change their structure, functionalities, and content for the individual user in real time. This is achieved by monitoring the user status, the system state, and the current situation that the user is facing. By using an adaption strategy (mostly rule based), the user interface is changed at run time.
- Adaptable User Interfaces [2]: This user interfaces are highly adjustable in terms of presentation of information, display of user interface and its components or user interaction/input concepts. The settings are usually stored in a user profile and the user is able to adjust those settings in advance, usually in a settings dialog. During runtime, in contrary to the adaptive user interfaces, these settings do not change.

According to Laive [3], methods for user interface adaptations can further be assigned to the following categories:

- Adaptable/Manual: the user manages the process and performs all actions
- Adaptable with system support/user selection: the user dominates the adaptation process and the system supports it
- Adaptive with user control/user approval: the system dominates the adaptation process under the supervision of the user. The system initiates the action and notifies the user about the alternative that he/she has to choose
- Adaptive/Fully adaptive: the whole process is managed by the system, which decides and implements the action based on the preferential model and the main uses

Adaptive user interfaces show great potential towards enhancing the usability and accessibility of computer systems. User tracking with state-of-the-art sensors could give estimations about the current user's status, and could trigger adequate system reactions based on that [4, 5].

However, the added adaptability for user interfaces to improve accessibility might have some unwanted side effects. For example, increasing the font size to address the vision impairment of a person might result in longer text passages and the need to scroll, which in turn results in increased attention and memory demands for the user. Therefore, providing extensive adaptability is a highly complex task, as side effects and conflicts are difficult to locate [10]. Another unwanted site effect of fully adaptive user interfaces is the inconsistency caused by the dynamic changes to the user profile, which is then reflected in the user interface. This is another drawback, as consistency across webpages is very important for people with cognitive disabilities and also addressed in Guideline 2.1: Predictable of the W3C Web Content Accessibility Guidelines (WCAG2.1) [8, 11].

This paper describes how user interface adaptations are realized within the Easy Reading framework. Easy Reading is a software tool that supports cognitive accessibility of web content and enables people with cognitive disabilities to better read, understand and use web pages. This is achieved through functionalities as:

- Adjustment of the layout and structure of webpages,
- Explanation/Annotation of web content with symbols, videos or pictures,
- Automatic/supported Modification of web content e.g. by translating it into plain language or easy2read.

Easy Reading has been designed as a cloud based solution, allowing people to interact with clients implemented as browser extension or mobile applications. Within the framework, user interfaces, user interaction and the provided help are adaptable and, to a certain extent, also adaptive for the individual user.

2 State of the Art

In recent years several research projects have been dealing with the creation of adaptive user interfaces for people with disabilities. Among those projects, prominent examples are GPII [6] or MyUI [7].

GPII allows the personalization of user interfaces, by the use of cross-platform user profiles for user interface settings, and rule-based and statistical approaches for matchmaking [14]. The architecture of the GPII was developed by the Cloud4all project uses an ontology of user needs and preferences that are directly linked to user settings [9]. The linking is done with a rule based matchmaker (RBMM) that matches user preferences and needs, with solutions available to the system and settings supported by the solution. The matchmaker results therefore in a fully configured solution on the specific system based on the individual preferences and needs of a user [6].

MyUI on the other hand was an EU funded project that enabled the generation of individualized user interfaces that would adapt to the individual users needs in real-time, based on a user profile and the actual device [10, 12, 13].

These approaches all work with a user profile that is usually stored online. Once the user logs in, the profile is downloaded and a mechanism uses this profile to create a dynamic configuration of software, assistive technology, user interface or the whole operating system for the individual user. Adaptations can only be made on features and software that are currently available on the actual device or software, and therefore the user experience might change on different devices. While this approach is sufficient for most users, it is problematic for people with cognitive disabilities, as consistency of user interfaces is very important for them [8]. Another drawback of this solution is that features must be installed on the device first, before they can be used and adapted, which might be another obstacle for people with cognitive disabilities.

3 Approach

The Easy Reading framework allows users to obtain assistance for difficult to cope with content on any webpage. This is done by cloud based software-clients that inject a dynamically generated user interface directly in the current page. By this users are able to trigger different forms of help provided by the framework. The result of the support is then rendered again directly in the webpage – allowing the user to stay at the original content and learning to cope with it in the future.

Figure 1 shows a screenshot of Easy Reading on a Wikipedia page. The user interface is dynamically injected on the right – the result of triggering an assistance tool provided by the framework is directly rendered within the web-page. In this case the help was an automatically created AAC¹-version of the second paragraph accomplished by a text analysis cloud service in combination with an AAC library.

Adaptations within the Easy Reading framework can be applied to the user interface, the help that is provided, the user interaction (how help is triggered) and finally

¹ https://en.wikipedia.org/wiki/Augmentative_and_alternative_communication.

Fig. 1. Easy Reading on a webpage

how the help is rendered and presented within the web-page. Similarly to existing approaches, these adaptations are based on a user profile that stores user preferences and abilities. Currently the user profile hosts the following support categories for the help provided by the framework:

- Text Support: Indicates whether and how the user needs help with text and content in general.
- Layout Support: How the layout of Websites should be displayed for the user.
- Reading Support: If and how the user needs support in reading text
- Symbol Support: Indicates if and how the user needs support with symbol language

In addition, the profile holds categories for triggering and displaying the provided help:

- Input Support: Stores the preferred way to triggering help and to select where on the web-page help is needed
- Output Support: Specifies the preferred way of rendering the help provided

Based on these categories, once the user logs in with his or her user profile, a dynamically optimized configuration is created for the individual user (see Fig. 2). Unlike other approaches, this configuration is not created locally, but in the cloud, and it also includes personalized user interfaces, personalized help and a personalized way of displaying the help. In this manner, clients within the Easy Reading framework do not host the code for any feature provided by the framework, as this is dynamically created for each user.

This is a big advantage over other architectures, as no matter from which device the user logs in, the user experience is always the same. Another advantage is that no additional software needs to be installed, as every feature is prepared in the cloud and downloaded during user login. Finally this enables learning and improving personalization of service provision cross different web pages and over time.

Fig. 2. User interfaces dynamically created and configured for the individual user

A drawback of the solution is however that it only works within a browser, while other solutions like GPII would also work across different applications or even operating systems. Here however each application has to be GPII-compliant and must implement its interfaces. Expanding the Easy Reading approach towards this broader application scenarios are considered as future challenges.

4 Update Strategies

As user skills, know-how and preferences change over time, the Easy Reading framework hosts a mechanism that automatically updates the preferences of the user profile based on user tracking and usage statistics (Fig. 3).

While the user is surfing the web, user interaction and tool usage is evaluated, and an updated profile is calculated. In addition, the system also hosts an additional optional user tracking component that creates an estimation on the users' current focus and detects and understands the situation the user faces (e.g. attention, stress, and confusion) at the moment. By this the additional feedback is created whether

- the user needs help for a part of the content,
- the help applied by the framework is accepted by the user
- the user has problems with the user interface or the user interaction required to trigger help of the framework

User tracking combines different sensors that feed into a software reasoner to calculate this estimation. Currently an eye-tracker that tracks the focus of the user on the web-page is used to detect cognitive load. Additionally, a smartwatch that detects heartbeats and heart rate variability is utilized to detect stress.

Based on this sensor data and the user interaction on the web-page, every hour the matchmaking component is triggered with the updated profile, resulting in a new dynamic configuration. Based on this a recommendation to add or remove functionality is triggered and presented via a dialog to the user.

Fig. 3. Adaptive workflow within the Easy Reading framework

Figure 4 shows such a recommendation dialog. If the user accepts the dialog, the updated profile is saved and the tool is added into the current user interface. On the other hand, if the user rejects the recommendation, the changes to the profile are reverted. User approval of any changes is of utmost importance, as consistency of user interfaces and user interaction must be preserved.

Fig. 4. Tool recommendation within the Easy Reading framework

5 Current Results and Further Work

Currently the system is able to make recommendations for different tools to simplify web content as well as for different user interfaces that the framework provides. In the future recommendations on changing user interaction to trigger tools and displaying help provided by the framework will be implemented.

Due to the COVID-19 outbreak large scale user tests were not possible. Preliminary tests with 8 end users showed that a purely adaptive user interface without any user approval is not appropriate for end users. On the other hand, most users were able to understand the current implementation with user approval. Once the COVID-19 situation allows it, more exhaustive user tests are planned.

Acknowledgments. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 780529).

References

1. Benyon, D.I.: System Adaptivity and the Modelling of Stereotypes. National Physical Laboratory, Division of Information Technology and Computing (1987)
2. Hook, K.: Evaluating the utility and usability of an adaptive hypermedia system. *Knowl. Based Syst.* **10**(5), 311–319 (1998)
3. Lavie, T., Meyer, J.: Benefits and costs of adaptive user interfaces. *Int. J. Hum. Comput. Stud.* **68**, 508–524 (2010). <https://doi.org/10.1016/j.ijhcs.2010.01.004>
4. Brusilovsky, P., Maybury, M.T.: From adaptive hypermedia to the adaptive web. *Commun. ACM* **45**(5), 31–33 (2002)
5. Peissner, M., Schuller, A., Spath, D.: A design patterns approach to adaptive user interfaces for users with special needs. In: Jacko, J.A. (ed.) *HCI 2011. LNCS*, vol. 6761, pp. 268–277. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21602-2_30
6. Loitsch, C., Weber, G., Kaklanis, N., et al.: A knowledge-based approach to user interface adaptation from preferences and for special needs. *User Model. User-Adap. Inter.* **27**, 445–491 (2017). <https://doi.org/10.1007/s11257-017-9196-z>
7. Peissner, M., Häbe, D., Janssen, D., Sellner, T.: MyUI: generating accessible user interfaces from multimodal design patterns. In: *Proceedings of the 2012 ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2012* (2012). <https://doi.org/10.1145/2305484.2305500>
8. WCAG2.1 – Web Content Accessibility Guidelines. <https://www.w3.org/WAI/standards-guidelines/wcag/>. Accessed 16 Apr 2020
9. Madrid, J., Peinado, I., Koutkias, V.: Cloud4all Priority applications and User Profile Ontology (D101.1). Public Deliverable of the Cloud4all Project (2012). <http://cloud4all.info/render/binarios.aspx?id=90>. Accessed 16 Apr 2020
10. Peissner, M., Edlin-White, R.: User control in adaptive user interfaces for accessibility. In: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) *INTERACT 2013. LNCS*, vol. 8117, pp. 623–640. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40483-2_44
11. Solovieva, T., Bock, J.: Monitoring for accessibility and university websites: meeting the needs of people with disabilities. *J. Postsecond. Educ. Disabil.* **27**(2), 113–127 (2014)
12. García, A., Sánchez, J., Sánchez, V., Hernández, J.A.: Integration of a regular application into a user interface adaptation engine in the MyUI project. In: Miesenberger, K., Karshmer, A., Penaz, P., Zagler, W. (eds.) *ICCHP 2012. LNCS*, vol. 7382, pp. 311–314. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31522-0_46

13. Peissner, M., Schuller, A., Ziegler, D., Knecht, C., Zimmermann, G.: Requirements for the successful market adoption of adaptive user interfaces for accessibility. In: Stephanidis, C., Antona, M. (eds.) UAHCI 2014. LNCS, vol. 8516, pp. 431–442. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07509-9_41
14. Zimmermann, G., Strobbe, C., Stiegler, A., Loitsch, C.: Global Public Inclusive Infrastructure (GPII) – Personalisierte Benutzerschnittstellen. i-com. **13**, 29–35 (2014). <https://doi.org/10.1515/icom.2014.0027>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.