

Patterns for User Interface Adaptations

Towards Runtime Adaptations for Improving the Usability of Web Forms for Elderly Users

Gottfried Zimmermann^(✉), Annkristin Stratmann, David Reeß, and Tobias Glaser

Responsive User Interface Experience Research Group, Stuttgart Media University,
Stuttgart, Germany
gzimmermann@acm.org, {as198, dr044, tg049}@hdm-stuttgart.de

Abstract. Websites and web applications that require user input via web forms can be a usability barrier for elderly users if not designed carefully. This issue is even compounded by a broad diversity of needs and preferences as observed in this group of users. In this paper, we report about a current study in which we prototypically implemented and empirically evaluated four exemplary patterns of user interface adaptation. These patterns allow for dynamic substitution and/or augmentation of user interface parts at runtime, with the goal of improving the individual usability for an elderly user in a specific use context. This approach could eventually lead to highly personalized web forms within GPII and URC enabled infrastructures.

Keywords: User interface adaptation · Personalized user interface · Web forms · Supplemental user interface resources · GPII · URC

1 Introduction

User interaction with Web forms can be challenging for users, in particular for older users¹ who are unfamiliar with entering data on the Web [1]. Web forms that are not designed with elderly users in mind often result in low usability for those users, causing high error rates, poor efficiency and a low user satisfaction.

Many websites require form input from their users. This applies in particular to public Internet services (e.g. online government services, online banking, online shops, online travel agencies) which are becoming more and more essential for the citizens. With an increasing portion of people belonging to the group of “elderly people” due to the demographic change, it is important to allow for their full participation in a digital society.

¹ The term “older users” is often used for people of age 50 + , but this is not consistent throughout literature. In this paper, the terms “older users”, “elderly users” and “seniors” are used as synonyms.

However, there are a vast number of websites that have not been designed with the needs of senior citizens in mind. Also, many web designers and developers are currently working on Internet services of the future, but do not have the knowledge and expertise to design them in a senior-friendly way. Is it possible to repair the existing online services with regard to their usability for elderly persons? Can we do so in a well-structured approach, driven by patterns of replacement and/or substitution, even at runtime?

Older adults are quite diverse in their digital interface preferences and needs [2]. The diversity is caused by factors such as their previous technical education and experiences, and their attitudes, but also by their abilities (in particular cognitive, motor, seeing, hearing) which may be affected by age [3]. Therefore, older users should not be understood as *one* coherent user group, as represented by one persona only.

A “one size fits all” approach would not be adequate for meeting the needs of elderly users. Therefore, the Global Public Inclusive Infrastructure (GPII) [4] promotes a “one size fits one” approach in which a user interface is adapted based on the needs and preferences of an individual user. Nevertheless, it is useful to investigate candidate patterns of user interface adaptation with regard to how many users could potentially benefit from them. In this paper, we focus on patterns that could be applied on Web forms at runtime to cater for “quick fixes” to improve the personal experience of a particular user.

The remainder of this paper is structured as follows: Sect. 2 provides an overview of existing guidelines on the design of web forms for older users. In Sect. 3, we briefly explain the concept of user interface adaptations at runtime for the purpose of improved usability. Section 4 proposes some exemplary patterns of user interface adaptations that could be applied on Web forms, and reports about empirical findings of a study on these patterns. Finally, we provide conclusions in Sect. 5, and relate the study’s results to the future of user interface adaptations.

2 User Interface Guidance on Web Forms for Older Adults

The American Association of Retired Persons (AARP) published a literature review on designing web sites for older adults in 2004 [3]. This literature review and some follow-up empirical research [5] resulted finally in 20 high-level heuristics (broken down into checklist items) for “understanding older adults as web users” [6]. Among them, the heuristics on using task-relevant and familiar images for buttons (checklist items 2.5 and 2.6), and on avoiding scrolling lists (checklist item 4.3) relate to our pattern on using push-buttons vs. drop-down menus (see Sect. 4.1). The heuristic on descriptive error messages (checklist item 7.1) contributes to our pattern on validation on submit vs. validation on input field (see Sect. 4.2). Also, the advice on making pages look clean and well organized as opposed to being cluttered or busy (checklist item 13.1) is relevant for our pattern on the display of help text (see Sect. 4.4).

Kurniawan and Zaphiris structured existing guidelines on web design for older people in eleven categories containing a total of 38 guidelines [7]. Among them, the guidelines on simple and meaningful icons (H2.3) and on avoiding pull-down menus (H3.4) are a direct input for our pattern on using drop-down menus vs. push-buttons (with graphical symbols) (see Sect. 4.1). The advice on simple and easy-to-follow error

messages is reflected in our pattern on validation on submit vs. validation on input field (see Sect. 4.2).

The National Institute on Aging in the United States has released their own set of guidelines for making websites “senior friendly” [8]. Among them, the guidance on using the same symbols and icons throughout a website (under the layout category) is relevant for our pattern on using drop-down menus vs. push-buttons (with graphical symbols) (see Sect. 4.1).

Pernice et al. [1] developed 106 design guidelines for “seniors citizens on the web”, based on quantitative research. Their advice on presenting error messages clearly (guidelines 34) relates to our pattern on validation on submit vs. validation on input field (see Sect. 4.2). The guideline on showing only pictures that are clear or can be easily zoomed in is reflected in our pattern on making images available in enlarged format (see Sect. 4.2). The advice on being forgiving of errors in forms is a direct input for our pattern on validation on submit vs. validation on input field (see Sect. 4.2). While Pernice et al. do not completely discourage designers from using drop-down menus in general, they remind designers of the difficulties that could come with them for older users (guideline 92). Considering their guidance on ensuring that images are easy to see (guideline 53), our pattern on using drop-down menus vs. push-buttons (with graphical symbols) can be seen as a natural application of these guidelines (see Sect. 4.1).

Almost all guidelines emphasize that their application will make websites easier to use for all users, not only for older users. In particular, Chadwick-Dias et al. [9] report about a study in which performance improved significantly for both older and younger users when a website was redesigned to accommodate the usability issues that older users had.

3 Dynamic User Interface Adaptations

We postulate that it is possible to improve usability of Web forms for elderly users by applying dynamic adaptations on its presentation, structure and content. The adaptations could be performed at runtime, based on previously identified replacement and augmentation patterns. For example, one of the replacement patterns we investigated is to replace a drop-down menu with text entries by a set of push-buttons with graphical symbols. We assumed that elderly users prefer the set of graphical push-buttons because here they see their choices more plainly, without having to click on a small button to open a list of text entries. The result of the investigation on this and other patterns is presented in Sect. 4.

In [10], a framework for the development of fully adaptive user interfaces is presented, consisting of the following six steps:

1. *User Interface Modeling.* An abstract user interface is modeled (development time).
2. *Default User Interface Design.* A default user interface is designed, based on the abstract user interface model (development time).
3. *Supplemental User Interface Design.* Supplemental user interfaces for specific contexts of use are designed, based on the abstract user interface model (between development time and runtime).
4. *Context of Use Instantiation.* At runtime, a context of use is identified, including concrete values for a user model, a platform model and an environment model.

5. *User Interface Accommodation (System-Driven)*. The system dynamically adapts its user interface to be tailored for the concrete context of use (at runtime). Dynamic adaptations can use the supplemental user interface components of step 3 (this is called *user interface integration*) or just “tweak” some aspects of the user interface, e.g. font size or mouse speed (this is called *user interface parameterization*).
6. *User Interface Customization (User-Driven)*. The user requests the system to adapt its user interface along pre-defined options. This can happen through user interface integration and/or user interface parameterization at runtime.

In this 6-step framework, patterns for improving the usability would be prepared in step 3 (supplemental user interface design), and would be applied either automatically as part of step 5 or – on user request – as part of step 6. In case of automatic (system-driven) pattern application (step 5), the system would first check the given context of use (step 4) and whether the application of the pattern is likely to improve the usability. This could potentially involve a matchmaker [11] for finding best adaptation candidates for the concrete context of use, and potential user interface adaptation patterns. These patterns could be provided as a pool of supplemental user interface resources (step 3) and possible parameters for user interface parameterization. In case the user requests the adaptation along a pre-defined pattern (step 6), this would be stored in the user model part of the context of use, so that the system would remember the user’s preferred pattern for the upcoming interaction sessions.

MyUI [12] is a pattern-based system for user interface adaptations. It automatically generates a user interface at runtime, based on an abstract user interface description borrowing elements from state machine diagrams (called *abstract application interaction model*). MyUI lacks a possibility for designers to hand-craft a user interface (step 2 of the 6-step framework) which is usually an important aspect for today’s design processes. Also, designers and developers are usually not familiar with the development of abstract user interface models based on state machines, and prefer imperative approaches. Nevertheless, MyUI’s adaptation patterns have influenced our thinking about dynamic adaptations for elderly Web users.

For other work related to user interface adaptations, refer to [10].

4 Exemplary Patterns of User Interface Adaptations

As described in Sect. 2, there are a large number of design guidelines available on how to make websites and web applications easy to use for older users. Many of these guidelines are supported by empirical data. However, we wanted to make the case for particular patterns of dynamic adaptations that could be run automatically to make web forms easier to use for an elderly individual. These should be applicable even when the author of the web form did not consider the needs and preferences of older users at all. Therefore, we decided to prototypically implement a selection of four substitution/augmentation patterns of which most are based on or related to existing guidelines (see Sect. 2). The prototype was implemented with Axure RP.²

² <http://www.axure.com/>.

In a study with desktop computers, we tested these patterns on acceptance for older adults. We had a total of 15 subjects (4 women, 11 men), aged between 64 and 84. We asked them for their experience with computers and the Internet. 10 subjects had already used computers in their workplace, 13 had used emails for communication, and 9 had already ordered a product in an online shop. During our observations, we noticed that 10 of the subjects used the tab key to navigate between input fields, which is an indication that they were quite familiar with the web and its forms.

Every subject got the same set of tasks:

1. Search for a specific product (sweater or t-shirt).
2. Put two pieces of the product in size “M” into the shopping cart.
3. Proceed with the payment process.
4. Enter name and address for delivery.
5. Pick a delivery method.
6. Pay with a credit card (a “faked” MasterCard was given to the user for this purpose).

In this set of tasks, we exposed the participants to four patterns of user interface adaptation, each coming in two versions (see subsections 4.1–4.4). For example, we had them order a sweater, whereby they had to choose the size from a drop-down menu (version A). After that, they had to order another sweater, this time selecting the size from a set of push-buttons with graphical symbols and letters for the different sizes (version B). We varied the order of version A and version B between subjects to counter-balance possible learning effects. At the end of the test, we collected feedback from the users on the different versions of the patterns.

The following subsections describe the patterns and the pertaining results of the empirical study.

4.1 Drop-Down Menu vs. Push-Buttons

In general, expandable menus that require a user action to open are often considered an issue for older users (see Sect. 2). This can be attributed to dexterity issues and cognitive difficulties. The menu choices are hidden and become only visible after clicking on a small button at the right of the drop-down menu field. Obviously, this is also an issue for users of touchscreen devices since it is hard to hit the “open menu” button with the finger. Also, if the user is not familiar with a drop-down menu, they may not know how to open it.

In our study, we contrasted drop-down menus with text items to sets of push-buttons, for selection of size and number of shopping items. In the choice of size, we also added the letter “S”, “M” or “L” on top of the symbol to make their meanings more obvious (see Fig. 1).

The seniors’ feedback was not completely uniform. Some commended the ease of recognizing and selecting the pictograms in version B, whereas others – being aware of their computer expertise – thought the pictograms were inappropriate and made them feel like alphabets or children. Also, some wanted to have a text input field for the number of items, so they would not be restricted to the values 1, 2 and 3.



Fig. 1. Version A (left) and version B (right) of the pattern “drop-down menu vs. push-buttons”, implemented as part of a product page. In version A, size and number of items must be selected from a drop-down menu. In version B, push-buttons are shown for size and number.

4.2 Make Images Available in Enlarged Format

In many online shops it is already customary to have high-resolution versions of images available upon demand (often triggered by clicking on a magnifier glass icon in the upper left of the image). While the image file may only be available in low resolution – if the author of the web page did not bother to provide different versions of the image – a system could always add an “enlarge image” function at runtime. Once activated, this function would display the image on a larger area by stretching (e.g. in a popup window, see Fig. 2). For many users – in particular those with low vision – the fact that the large image version is not rendered in high resolution would be a minor problem, as long as the image is enlarged.

Most participants found that the possibility of enlarging the product images improves the website’s ease of use. Many senior users were already familiar with this functionality due to its widespread use in online shops. Some even wanted to have multiple views on the items, e.g. from different perspectives, or have selected areas of the image magnified.

4.3 Validation on Submit vs. Validation on Input Field

Clear and descriptive error messages are an important aspect of senior-friendly websites (see Sect. 2). A related aspect is the timing of warnings in case of invalid form input. The user may be alerted to validation errors only after they have filled out and submitted the complete form (pattern version A), or they may get a message right away when they have left an input field with invalid content (version B). Figure 3 shows our prototypical implementation of this pattern, in both versions.

All participants were either pleased with the validation of input fields upon focus change (version B), or had a neutral opinion. One user appreciated the direct feedback – or better in this case the absence of a negative feedback – as a kind of reward for a doing a good job. The validation upon submit (version A) caused confusion for some elderly users since they did not immediately understand what the problem was and what they had to do to be able to move on.



Fig. 2. Version B of the pattern “make images available in enlarged format“. On every product image, a magnifier class with a ‘+’ sign is superimposed in the upper left corner. If the user clicks on it, a large image version is displayed in a popup. Note that version A (no screenshot included) has no magnifier glass on images, and the user cannot request an enlarged image version.

<p>Vorname</p> <input type="text"/> <input type="text"/>	<p>?</p> <p>Bitte geben Sie einen Vornamen an.</p>	<p>Vorname</p> <input type="text"/> <input type="text"/>	<p>Bitte geben Sie einen Vornamen an.</p>
<p>Nachname</p> <input type="text" value="Zimmermann"/> <input type="text"/>	<p>?</p>	<p>Nachname</p> <input type="text"/> <input type="text"/>	<p>Tragen Sie Ihren Nachnamen ein</p>
<p>Postleitzahl</p> <input type="text" value="70569"/> <input type="text"/>	<p>?</p>	<p>Postleitzahl</p> <input type="text"/> <input type="text"/>	

Fig. 3. Version A (left) and version B (right) of the pattern “Validation on submit vs. validation on input field“. In both versions, the user did not enter their first name. In version A, the red validation message is only shown after the user has pressed the “submit” button (not included in the screen shot). In version B, the user is alerted to the error as soon as they set the focus to the next field.

4.4 Display of Help Text

Help texts for web forms can be presented to the user in various ways. Some websites display hints and help texts openly, others on the press of a button. Recently, some websites have provided hints as “placeholders” in input fields, but unfortunately a placeholder text disappears as soon as the user sets the focus in the field. On the other hand, hints and help texts may clutter the screen and distract the user (see Sect. 2).

In our fourth pattern, we implemented two versions of help text display for input fields. In version A, the help text appears below the input field when the user clicks on a button with a question mark (‘?’). The button is placed to the right of the input field.

The help text disappears again when the user leaves the input field. In version B, the help text is shown automatically when the user sets focus to the input field, and disappears when the user leaves the input field (see Fig. 4).

Most users did not notice the help texts, neither for version A nor for version B. When asked for feedback, they deemed the input for name and address fields as trivial and not requiring any hints beyond the label. However, they appreciated help for more complex input fields such as the expiration date of the credit card. Here, they wished a permanent display of help text that would not disappear when moving away from the input field (as opposed to both versions implemented in our prototype), e.g. in the form of a long label.

4.5 General Observations

Aside from the assessment of the four patterns for user interface adaptation, the study brought about some other findings that could potentially be used in automatic user interface adaptations.

In general, participants had problems in recognizing that they needed to scroll vertically to get to the screen elements “below the fold”. This problem is a common finding and reflected in various guidelines (see Sect. 2). An adaptation system could automatically break up one page into multiple pages to avoid the need for scrolling.

A related problem occurs when – upon a submit operation – error messages are displayed in a part of the screen that is currently not visible due to scrolling. This could be avoided by automatic scrolling or – even better – by validating upon leaving an input field rather than upon submit (cf. Sect. 4.3).

When filling out web forms, many participants used either all-capital characters or all-small characters in their typing. An adaptation system should be tolerant to this practice, and could further automatically correct the character case where appropriate.

The division of first name and surname into two input fields was unnatural for some participants. They entered their full name into the field for the first name, and noticed their mistake when they got to the next input field (requesting the surname). The same occurred with the input fields for street name and house number, and with the input fields for the expiration date of the credit card (month and year). An adaptation system could recognize these user input mistakes and automatically split the input and distribute its pieces into the appropriate fields.

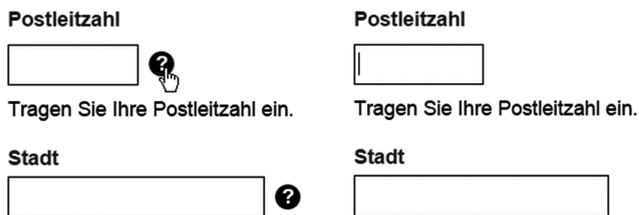


Fig. 4. Version A (left) and version B (right) of the pattern “display of help text“. In both versions, input fields for the ZIP code and the city are displayed, and a hint is shown under the first input field: “Enter your ZIP code“. In version A, the hint is shown upon pressing a ‘?’ button. In version B, the hint is always shown when the focus is on the input field.

On typing their email address, many seniors typed a comma (‘,’) instead of a dot (‘.’) due to vision and dexterity problems. This mistake could also be automatically corrected.

For the credit card, some participants entered its number in blocks of 4 digits, with spaces between the blocks. In fact, this allows for easier reading and double-checking of the number. An adaptation system could offer separate input fields for each block, and automatically jump to the next block when a block is full.

A frequent issue for the senior users was the entry of the 3-digit card validation code which they were not familiar with. An adaptation system could provide additional guidance for this input field on request, for example by a photo explaining where this code can be found on a real credit card.

Some participants did not have an email address. Therefore, an alternative contact for clients should be offered (e.g. a phone number), and the email address field should not be a mandatory field. While this problem cannot be easily fixed at runtime, it could be part of guidelines for web authors.

5 Conclusions and Outlook

The findings of this paper relate to the concepts of the Global Public Inclusive Infrastructure (GPII) [4], and the Universal Remote Console (URC) framework [13]. We have identified four exemplary patterns that can facilitate the dynamic personalization of Web forms at runtime. This approach fits well within the framework of fully context-sensitive user interfaces [10], in particular for the steps on “user interface accommodation (system-driven)” (step 5) and on “user interface customization (user-driven)” (step 6) (cf. Sect. 3). As always, the user should be in control over the adaptations performed, even though the system may anticipate and propose specific adaptation patterns.

It is important to note that for many adaptations at runtime, the author does not need to be involved. In fact, all exemplary adaptation patterns presented in this paper could be achieved in a fully automated fashion and facilitated by third-party authors providing supplemental user interface resources before runtime. Supplemental resources for personalization can be deployed through an online resource server [14] if the basic structure and abstract content of the user interface is known before runtime. Of course, the provision of supplemental user interface resources has significant security implications. Care needs to be taken for such a framework to prevent from malware sneaking in through the door of “supplemental resources”.

None of the patterns that we investigated had a clearly superior version that would be preferred by a majority of users. Instead, this study confirms the findings of others that elderly users are a rather heterogeneous audience with a broad diversity of needs and preferences (see Sect. 2). The group of elderly users makes a strong case for the concept of personalized adaptations on user interfaces, as facilitated by the GPII and URC frameworks. Therefore, the question resulting from this study is not which version to implement for an adaptation pattern, but rather to allow for all versions to be available as possible adaptations for a user interface. That way a suitable version can be selected and instantiated at runtime, driven by the actual context of use, including the user’s personal needs and preferences.

Acknowledgments. We thank Dr. Ulrich Schloz for his kind support in facilitating the user tests with seniors from the association “Senioren Online” in Reichenbach, and from the association “Arbeitsgemeinschaft Mikroprozessor und Minicomputer” in Stuttgart. We also thank the nursing home “Seniorenzentrum am Rosengarten” in Bondorf for their support and participation in the user tests.

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2011) under the grants 289016 (Cloud4all) and 610510 (Prosperity4All). This publication reflects only the authors’ views and the European Union is not liable for any use that may be made of the information contained herein.

References

1. Pernice, K., Estes, J., Nielsen, J.: Senior Citizens on the Web, 2nd edn., Research report by Nielsen Norman Group. <http://www.nngroup.com/reports/senior-citizens-on-the-web/>. Accessed 2013
2. Gregor, P., Newell, A.F., Zajicek, M.: Designing for dynamic diversity: interfaces for older people. In: Proceedings of the Fifth International ACM Conference on Assistive Technologies, pp. 151–156. ACM, New York. doi:10.1145/638249.638277 (2002)
3. Redish, J., Chisnell, D.: Designing web sites for older adults: a review of recent research, AARP. http://assets.aarp.org/www.aarp.org/_articles/research/oww/AARP-LitReview2004.pdf. Accessed 14 December 2004
4. Vanderheiden, G., Treviranus, J., Markus, K., Clark, C., Basman, A.: The global public inclusive infrastructure, Cloud4all and Prosperity4all. Assistive Technology: From Research to Practice, vol. 33, pp. 417–422. Presented at the AAATE 2013. IOS Press Ebooks, Villamoura, Portugal. doi:10.3233/978-1-61499-304-9-417 (2013)
5. Chisnell, D., Redish, J.: Designing web sites for older adults: expert review of usability for older adults at 50 web sites, AARP. <http://www.redish.net/images/stories/PDF/AARP-50Sites.pdf>. Accessed 1 February 2005
6. Chisnell, D., Redish, J., Lee, A.: New heuristics for understanding older adults as web users. Tech. Commun. 53(1), 39–59 (2006)
7. Kurniawan, S., Zaphiris, P.: Research-derived web design guidelines for older people. In: Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 129–135. ACM, New York. doi:10.1145/1090785.1090810 (2005)
8. NIA: Making your website senior friendly. National Institute on Aging, March 2009. <http://www.nia.nih.gov/health/publication/making-your-website-senior-friendly>. Accessed 15 August 2014
9. Chadwick-Dias, A., McNulty, M., Tullis, T.: Web usability and age: how design changes can improve performance. In: Proceedings of the 2003 Conference on Universal Usability, pp. 30–37. ACM, New York. doi:10.1145/957205.957212 (2003)
10. Zimmermann, G., Vanderheiden, G.C., Strobbe, C.: Towards deep adaptivity – a framework for the development of fully context-sensitive user interfaces. In: Stephanidis, C., Antona, M. (eds.) Universal Access in Human-Computer Interaction Design and Development Methods for Universal Access, pp. 299–310. Springer International Publishing, Heidelberg (2014)
11. Loitsch, C., Stiegler, A., Strobbe, C., Tzovaras, D., Votis, K., Weber, G., Zimmermann, G.: Improving Accessibility by Matching User Needs and Preferences. In: Assistive Technology From Research to Practice, pp. 1357–1365. IOS Press, Vilamoura. doi:10.3233/978-1-61499-304-9-1357 (2013)

12. Peissner, M., Häbe, D., Schuller, A.: Adaptation concept and multimodal user interface patterns repository, FHG. http://myui.eu/deliverables/MyUI_D2-2_final.pdf. Accessed 2011
13. Zimmermann, G., Vanderheiden, G., Ma, M., Gandy, M., Trewin, S., Laskowski, S., Walker, M.: Universal remote console standard - toward natural user interaction in ambient intelligence. In: Extended Abstracts of the 2004 Conference on Human Factors and Computing Systems - CHI 2004, p. 1608. ACM Press, New York. doi:[10.1145/985921.986169](https://doi.org/10.1145/985921.986169) (2004)
14. Zimmermann, G., Wassermann, B.: Why we need a user interface resource server for intelligent environments. In: Workshops Proceedings of the 5th International Conference on Intelligent Environments, vol. 4, pp. 209–216. IOS Press. doi:[10.3233/978-1-60750-056-8-209](https://doi.org/10.3233/978-1-60750-056-8-209) (2009)