Experiential Adaptation to Provide User-Centered Web Content Personalization

Silvia Mirri, Catia Prandi, Paola Salomoni
Department of Computer Science and Engineering
University of Bologna
Bologna, Italy
{silvia.mirri, catia.prandi2, paola.salomoni}@unibo.it

Abstract—**Personalization of pages on the basis of each specific users' need is a key factor in navigating the Web, in particular for those users who browse under specific conditions. This paper proposes to track user's behavior and to gain his/her effective needs by means of machine learning concepts. Our goal is to provide a system which improves Web legibility and readability, customizing typographic characteristics of pages. The system lets users adapt Web pages and learns how to apply automatic personalization. This can be of great benefit both for users with reading-related disabilities and for users accessing with non-conventional devices.**

Keywords-Content Adaptation; Web Personalization; Profiling Users; Maching learning.

I. INTRODUCTION

Personalization is a key feature in providing services and content which are effective for users. Self-adaptation of Web pages is strongly overworked by Web 2.0 applications. Nevertheless, this issue is not usually exploited so as to meet specific user's needs, making the Web pages personalization (both in terms of content and shape) an interesting and unsolved challenge [8, 14, 16].

In this field, a more user-centered approach is necessary to customize Web page elements, adapting their shape with the aim of meeting each single user's needs. Such kind of approach can have a strong impact, in particular for those users with some reading-related disabilities (i.e., aging people, people with dyslexia, people with low vision, users with color blindness, etc.). Moreover, this approach can make the Web content more accessible even for those users who are equipped with devices with different capabilities (i.e., different screen sizes, different interactions systems), such as tablets, smartphones, smart TVs, etc. In these contexts, both readability and legibility are affected by Web pages characteristics [5, 12], users' abilities [4, 9] and device capabilities.

In order to make Web pages (or just some parts of them, such as paragraphs, headings, links, tables, etc.) more legible and readable to users with disabilities and to users who exploit different devices, transcoding and adaptation activities can be performed. Transcoding and adaptation have been strongly exploited in research field and they are usually driven by categorizing device capabilities and users' needs [1]. In this paper, we propose an innovative Web pages adaptation system based on the concept of experience-based transcoding, called "experiential transcoding" [2]. Comparing with more traditional forms of content transcoding and adaptation, the main advantage of experiential transcoding is that it is strongly user-centered. Furthermore, it applies techniques and mechanisms which adapt content on the basis of users' experience, by understanding and predicting it [13].

The main aim of this work is to improve Web pages legibility and readability by adapting some characteristics (such as font size, font face, luminance contrasts, and so on) according to users' preferences and needs. To reach this goal we have designed a system which lets users adapt Web documents, tracking users' behavior (on the basis of the device in use), so as to learn and model their preferences and to automatically provide the best adaptation, tailored for each user, predicting his/her needs.

In order to understand user's experience and to learn user's preferences, we have used a machine learning mechanism, the Reinforcement Learning [15] one, based on the idea of reward/punishment. Thanks to this, we can build and feed a user's profile which models his/her preferences in terms of Web page characteristics affecting his/her reading ability (both in positive and in negative ways). Thus, the Web content adaptation will be more user oriented, meeting each single user's need.

In this paper, we present the system we have designed and a prototype we have developed which can adapt HTML pages. The main advantage of our system is the deep and detailed user's profile we can gain: the system tracks the user's behavior and continuously updates his/her profile, (the more the user exploits it and the more the profile will be accurate). This lets the system take into account users' preferences and needs, even when they change. Moreover, this means that users are not categorized according their disabilities, but the system can provide Web content personalization for each single user. Without such adaptations a user's need can conflict with general best practice, or can conflict with another user's need. For instance, many people with dyslexia and other reading impairments need low contrast between text color and background color [9], while many people with low vision and many people with declining eyesight due to ageing need high contrast. On the contrary of other similar works, in which the entire content is customized according to user's preferences and needs [7, 8, 9, 11, 16], our system adapts only those elements in the Web page which present some characteristics that can affect the user's reading, without distorting the whole page and its layout. Currently the prototype is under a testing phase by simulating users with low vision and dyslexia. An evaluation campaign with real users will be conducted next months: this will report users' appreciation, the ease of use of the prototype, and the validity of our approach.

The remainder of the paper is organized as follows. Section 2 presents the system structure and it describes in details each system modules: the users profiling, the users' preferences learning and the adaptation ones. Section 3 shows the prototype we have developed. Finally, Section 4 concludes the paper presenting further work.

II. System Structure

In this section, we are going to describe our system structure, which includes: the users profiling module, the users' preferences learning module and the adaptation module, as depicted in Figure 1. Each module is described in details in the following subsections.

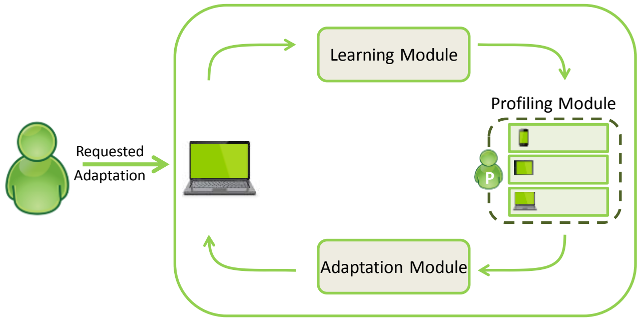


Figure 1. The whole system.

A. Users Profiling Module

Profiling users is an activity which can be useful in several contexts and applications, such as user interfaces and Web applications (i.e., recommendation system in e-commerce Web sites, personal data in social networks or in search engines filters, and so on). Data gathered in users' profiles can range from personal data to contextual conditions, from user's skills to his/her personal preferences [10]. Such data can be collected in different ways, for instance the user can explicitly declare them or they can be learnt by the system as the user exploits it.

In our system, the user profile is a collections of typographic characteristics gathered by the system (as described in the following Subsection 2.B), on the basis of adaptations (see Subsection 2.C) the user has requested in so as to improve Web content legibility and readability. Such a profile is shared among all the devices the user exploits and it takes into account different user's needs according to capabilities of the device in use (i.e., different display size, etc.).

When the user exploits the system and asks for adaptations, our system computes automatic adaptations and related reward/punishment values, tracking the user's behavior. Periodically, the system updates the user's profile, adding new characteristics, updating reward/punishment values for the already existing ones or adding new characteristic values as requested by the user.

The system provides adaptations as state changes: from the original one (let us call it A) to the adapted one (let us call it B), according to each user's request. Let us use the following formalism to indicate the state change from the original state A to the adapted state B: A→B.

Characteristics of state A (which are substituted by characteristics of state B) are the ones the user has discarded, while characteristics of state B are the ones the user has chosen. Hence, the system learns all the Web page characteristics which affect user's reading ability (both in positive and in negative ways). Obviously, the user's profile will be the more accurate the more the user asks for adaptations. As described in the following subsection, the system punishes discarded characteristics, while it rewards characteristics the user has chosen through the requested adaptations.

We have designed an xml-based profile which is structured in different parts, according to the devices the user exploits. In each of these parts, the system stores typographic characteristics (as tags) the user has preferred or the user has discarded, the related value (as the "**v**" attribute) and a number (as the "**w**" attribute). Such a number states the reward/punishment related to the adaptations asked by the user.

In fact, the "**w**" value varies according to user's behaviors and in particular:

- if "**w**" is < **0**: the user has discarded the characteristic with the related "**v**" value;
- if "**w**" is > **0**: the user has asked for the characteristic with the related "**v**" value;
- if "**w**" is = **0**: the characteristic with that "**v**" value has obtained the same quantity of rewards and punishments.

The absence of a characteristic or of a specific "**v**" value in the user's profile, means that the user has never asked for such characteristic adaptation or he/she has never discarded or chosen such a "**v**" value.

Hence, the profile is composed by a set of Web page characteristics the user has adapted and by the related values he/she has asked and discarded, grouped by the devices the user exploits. For each device, the profile stores the type, an id value, the display width and the display height (in terms of pixel). The device in use is deduced by the HTTP request and its capabilities are collected from repositories [17].

The Web page characteristics our system takes into account are related to text and to content features which can affect both legibility and readability [5, 12]. Typographic characteristics are related to fonts (i.e., face, size, style), spaces (i.e., word and letter spacing, margins, line height, alignment) and colors (i.e., background, foreground and luminance ratio) [4], while content characteristics are related to acronyms, abbreviations and foreign words or sentences.

Figure 2 shows a fragment of a user's profile, as an example. Web page characteristics are grouped by a tablet device. In such an example, the user has asked (among all the adaptations): Arial as font face, 18 as font size, and 1.5 as line height. In the same example, the user has discarded: Times new roman as font face, 9 as font size and 1 as line height, while he/she has used the tablet.

```
…
<device type=″tablet″ id=″2″ display_width=″″
display_height=″″>
        …
        <font_face family=″sanserif″ w=″5″
          v=″arial″/>
        <font_face family=″serif″ w=″-2″ v=″times
        new roman″/>
        <font_face family=″monospace″ w=″-5″
        v=″courier new″/>
        …
        <font_size w=″8″ v=″18″/>
        <font_size w=″-5″ v=″9″/>
        …
        <line_height w=″5″ v=″1.5″/>
        <line_height w=″-2″ v=″1″/>
        …
</device>
…
```

Figure 2. A fragment of a user’s profile.

B. *Users’ Preferences Learning Module*

In order to learn users’ preferences, our system has been modeled by using the Reinforcement Learning concept. The learning algorithm adopted is the Q-learning algorithm [15]. Q-learning is a popular Reinforcement Learning algorithm which works by estimating the values of state-action pairs. The value *Q(s, a)* is defined to be the expected discounted sum of future payoffs obtained by taking action *a* from state *s* and following an optimal policy thereafter. Once these values have been learned, the optimal action from any state is the one with the highest Q-value. Q-learning works by successively improving its evaluations of the quality of particular actions at particular states. The system learns the Q-values as the user exploits it. Then the system uses the learnt Q-value for making better decision about possible adaptations when the user requests a Web page. Q-values can thus provide estimation of how successful that action might be. The Q-learning algorithm has been already used in several works, where systems have been design to provide customizations according to learnt users’ preferences [11].

Our work focuses on directly interacting with the user to learn his/her preferences. In fact, the user contributes by helping the system which tracks the user’s behavior to adaptations. In particular, the system obtains some feedbacks from the user, in the form of reward or penalty.

This problem has to be formulated as a Reinforcement Learning problem. In order to do this, we have to set up states, actions and rewards. States represents sets of Web page characteristics (font face, font size, word spacing, line height, acronyms expansion, and so on). Actions are the adaptations explicitly requested by users, the adaptations proposed by the system and the adaptations the system automatically performs. Rewards are related to user’s behavior: if the user explicitly applies an adaptation or if the user accepts an adaptation the system proposes, then the reward for the chosen characteristics is +1. If the user rejects an adaptation the system automatically performs then the rewards is -1. A -1 reward is assigned also to those discarded Web page characteristics as well.

Hence the whole system works in the following way:

1. when the user opens up a Web page, the system parses its characteristics, taking into account the user’s profile.
2. If there are some characteristics the user has discarded (with a negative “**w**” value, state A), then the system computes if automatically adapting such characteristics, substituting the original values with the ones the user prefers (with the highest “**w**” value, state B) or just proposing such adaptations, changing the state from A to B (A→B).
3. Then the user exploits the Web page with a specific set of characteristics (state B): font size and face, spacing, alignment, background and foreground colors, acronyms, and so on.
4. The system computes reward/penalty values tracking user’s behavior, according to his/her feedbacks:
    - If the user ignores the adaptations the system has automatically performed (hence the user implicitly accepts state B characteristics), then the reward is +1. Else, if the user rejects such adaptations then the reward is -1.
    - If the user explicitly accepts the adaptations the system has proposed (characteristics in state B) then the reward is +1. Else, the reward is -1.
    - If the user applies a certain adaptation to a certain set of Web page characteristics, changing from original A state to the adapted B one (A→B) then the system assigns +1, as a reward, to the requested characteristics (in state B) and -1 to the discarded ones (in state A).
    - Unchanged characteristics in state A and in state B receive no reward.
5. Updated rewards and/or Web page characteristics adapted for the first time are stored into the user’s profile.

As an example, let us take into account a user who exploits his/her PC and asks for a paragraph font size increase in a Web page, from 10 to 18 pixel. The system assigns +1 as reward to font size 18 (B state) and -1 as reward to font size 8 (A state).

After updating the user’s profile, when the user opens up another page, the system parses the characteristics and finds a paragraph with font size 10. Such a characteristic is stored in the profile with a negative “**w**” value (let us consider the user’s profile depicted in Figure 2). In this example, we can have the following system behavior:

- if **`t < w < 0`**: the system proposes to adapt such a characteristic by substituting it with the “**v**” value with the highest “**w**” (font size 18, with “**w=7**”) only on the mousehover event. The user can accept such a proposal, then the system assigns a +1 reward to font size 18 (the new "**w**" value will be 8) and a -1 reward to font size 10 (its new "**w**" value will be -5), else the system assigns a -1 reward to font size 18 (its new "**w**" value will be 6);

- if **w < t**: the system automatically adapts such a characteristic by substituting it with the characteristics with the highest "**w**" value (again font size 18, with "**w=7**"). If the user rejects such an adaptation, then the system assigns a -1 reward to font size 18 (the new "**w**" value will be 6), else the system assigns a +1 reward to font size 18 (the new "**w**" value will be 8) and a -1 reward to font size 10 (its new "**w**" value will be -5);

where **t** is a specific threshold. Such a threshold is a negative integer value which can be differently set for each characteristic. At the moment, we are conducting a testing phase to define such thresholds for the most common characteristics.

In this example, the system proposes the same adaptation even when paragraphs with smaller font size are found in the Web page. For instance, the system can propose adaptation to font size 18 even for paragraphs with font size 8. More generally, the same consideration is taken into account also for those characteristics with numeric values, such as luminance ratio, color contrasts, word and letter spacing, line height, etc. Hence, if the user has discarded a characteristic with a specific numeric value, the system learns to adapt such characteristic with worse values.

C. Adaptation Module

Content adaptation is the action of transcoding or transforming content so as to meet users' preferences and needs, even according to the device he/she is using.

In our system, we provide content adaptations with the aim of improving Web pages readability and legibility. In our system, the adaptation process is in charge locally, on the client side. The system can locally decide and employ the most appropriate adaptations, according to user's profile, which is fed on the basis of the user's behavior.

The aim of the whole work is to adapt any kind of markup document, such as LaTeX, PDF, RTF documents, etc. But currently the system works on explicit and descriptive markup documents, in particular HTML pages. Our system performs adaptations by changing tags, attributes and related values: the system injects new tags or attributes and/or it substitutes original tags or attributes value with the customized ones, changing markups.

When a user opens up a Web page, then the system parses the DOM and the related style rules, by considering those characteristics with a negative "**w**" value in the user's profile. If such characteristics are found in the page, then the system decides if automatically adapting them (if the "**w**" value is less than a threshold "**t**") or if just proposing to the user such an adaptation (if the "**w**" value is greater than "**t**"). The discarded characteristics are substituted by the "**v**" values with the highest "**w**": since the user has chosen them the most (to improve his/her reading ability), so we can assume these are his/her preferences in terms of such characteristics.

For instance, let us consider the user's profile depicted in Figure 2. If the user opens up a Web page and the system finds a paragraph written in Times New Roman (with "**w**" equal to -7), then the system proposes an adaptation from Times New Roman to Arial as font face. The system chooses Arial instead of Verdana, because Arial "**w**" value is higher than Verdana one.

A list of adaptations the system can employ is the following one:

- *Zooming font size*: the system can increase font size by a specific percentage or unit.
- *Changing font face*: the system can substitute the original font face with another one.
- *Changing font style*: the system can set a specific style (i.e., italic, underline or normal) to the text.
- *Changing spacing*: the system can change spacing-related attributes (i.e., letter spacing, word spacing, line-height, margins, paddings, etc.).
- *Changing text alignment*: the system can set users' favorite text alignment (left, right, center or justify).
- *Enhancing luminance ratio*: the system can increase the luminance ratio. The system keeps the same background color and computes a new foreground color, so as to enhance the luminance contrast ratio.
- *Changing background and/or foreground colors*: the system can set different background and/or foreground colors, according to users' choice.
- *Language translation*: the system can substitute original words and/or sentences in translated ones.
- *Acronyms expansion*: the system can substitute acronyms with their related expansions.

III. Prototype

We have designed and developed a prototype of our system which adapts HTML documents. Such a prototype has been implemented as a Firefox extension. Users can activate a contextual menu to set the preferred adaptations on an HTML page. Then the system performs such adaptations by suitably changing the HTML and/or the CSS code of the page, on the client-side. In the meanwhile, the system tracks users' behaviors with the aim of learning their preferences and then automatically applying or proposing suitable adaptations.

This prototype has been implemented as a Firefox add-on by means of Mozilla SDK [6]. Scripts have been created so as to:

- provide an ad-hoc contextual menu, letting users choose among a sub-set of available adaptations (as shown in the screenshot depicted in Figure 3): changing font face changing font size, and changing background and foreground colors.

- Adequately modify the DOM of the HTML page, injecting new attributes or changing values for the already existing ones. In particular, the prototype can add "**`style`**" attributes with CSS rules to the element tag or can change the existing CSS rules values, according to the adaptations the user has requested. This way, thanks to CSS cascading feature, customized values of inline rules override the same ones eventually declared in external or internal CSSs.
- Add scripts to create and activate pop-ups: when an adaptation is just proposed to users, this is triggered on the mousehover event, by means of AJAX scripts.

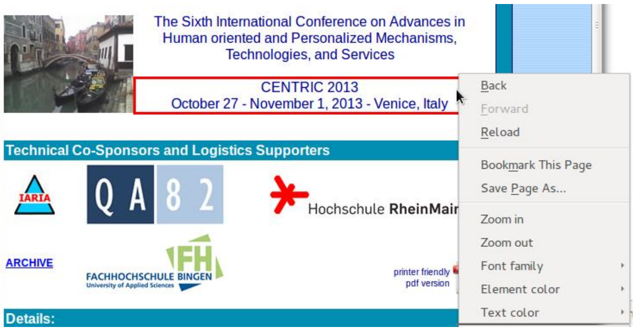
Currently, the prototype has been tested on laptops (equipped with different operating systems) and on Samsung Galaxy Tab 2 devices, equipped with Firefox browser. More browsers extensions are needed and are under development (i.e., for Chrome, Internet Explorer, etc.), so as to provide a wider and more complete system.

Figure 3 shows a screenshot of a Web page with the contextual menu of the prototype we have developed: after activating such a menu, it is possible to zoom in, to zoom out, to change the font family, to change the background and foreground colors of the specific element the user has chosen (which is highlighted by means of a colored border).

Figure 4 depicts a screenshot (of the same Web page in Figure 3), when an adaptation is proposed to the user: a pop-up is activated only on the mousehover event on the element the system proposes to adapt. The user can accept or reject the proposed adaptation, providing a feedback and letting the system learn about his/her preferences.

Figure 5 shows a screenshot (of the same Web page in Figures 3 and 4), when the chosen adaptation is automatically performed. In particular, the text in the chosen element has been increased from 16px to 24px, by means of zoom in adaptation. Screenshots in Figure 3, 4 and 5 have been taken from a PC equipped with Ubuntu 13.04 and Firefox 21.0.

A synchronization mechanism is still under development: synchronization among different devices used by the same user is needed so as to let the user enjoy customized Web pages on the basis of his/her preferences and of different device capabilities. Figure 6 shows the whole system architecture. Currently, each user stores his/her profile, structured in device-related profile sections (as reported in subsection 2.A), on each device he/she is using. Updates to the user's profile are locally stored (on the device in use) and a synchronization mechanism is needed to spread such updates on the other copies of the profile. The different copies of the profile have to be periodically synchronized. All the profiles will be stored on the server side. The copy of the profile stored on the server will be used in case of new devices associated to the same user: when a new device joins the system, the user's profile will be downloaded from the server and a new profile section will be added to all the copies of the profile, via the synchronization mechanism.

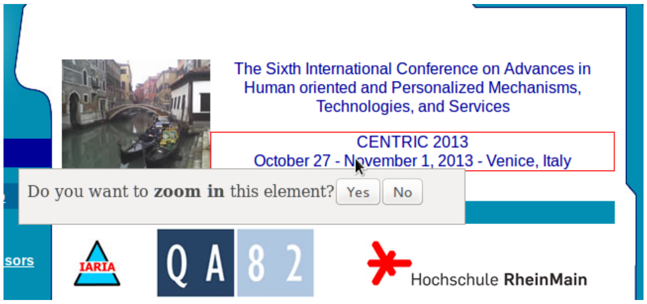

The Sixth International Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services

CENTRIC 2013
October 27 - November 1, 2013 - Venice, Italy

Technical Co-Sponsors and Logistics Supporters

Details:

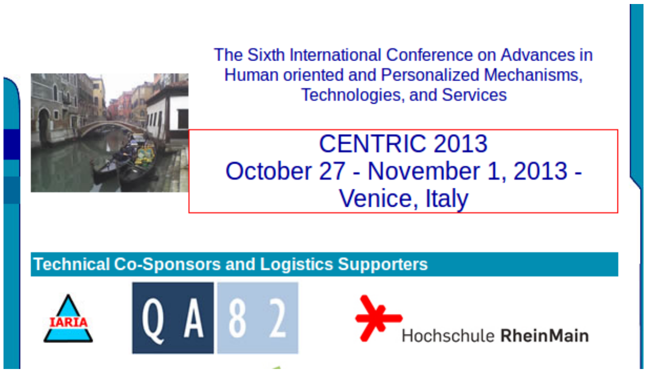
Figure 3. A screenshot showing our contextual menu.


The Sixth International Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services

CENTRIC 2013
October 27 - November 1, 2013 - Venice, Italy

Do you want to **zoom in** this element?

Figure 4. A screenshot showing an adaptation proposal.


The Sixth International Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services

CENTRIC 2013
October 27 - November 1, 2013 - Venice, Italy

Technical Co-Sponsors and Logistics Supporters

Figure 5. A screenshot showing an adapted element.

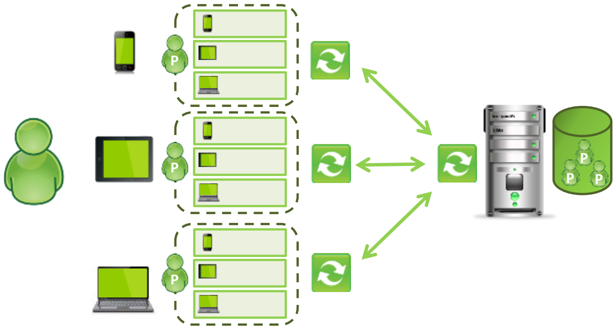


Figure 6. System Architecture.

IV. Conclusion and Future Work

Our work on improving Web pages legibility and readability by means of experiential transcoding (learning user's profile by tracking his/her behavior) is still under development, however we have presented in this paper the system we have designed and a prototype we have developed. The goal of our system is to adapt Web page characteristics (such as font size, font family, colors, luminance contrasts and so on) according to users' preferences and needs. The system we have presented lets users adapt such characteristics and tracks users' behavior with the aim of learning their preferences, so as to automatically provide the best adaptations, tailored for each single user, even on the basis of the exploited devices (PC, tablet, smartphone, smart TV, etc.).

Further work is needed to develop extensions and/or adds-on for the most commonly-used browsers (i.e., Chrome, Internet Explorer, etc.) and for other kinds of documents viewers and readers, letting the system adapt not only HTML pages, but also other markup documents (i.e., LaTeX ones). Moreover, further investigation is needed so as to adapt also PDF documents: some researches have been conducted studies on how to customize text characteristics on tagged PDF documents [3]. In order to provide a wider range of adaptable documents, these findings will be taken into account in our future work.

A system testing phase is ongoing. This is important to define suitable thresholds `t` for each document characteristic (in order to support the system in deciding when automatically adapt or just propose a specific adaptation). Finally, a user testing phase is needed and it would involve several users with different preferences and specific needs (i.e., aging people, users with low vision, users with dyslexia, users with color blindness, etc.) equipped with different devices.

ACKNOWLEDGMENT

The authors wish to thank Giulio Cirnigliaro and Luca Spadazzi for their precious support.

REFERENCES

[1] C. Asakawa and H. Takagi, "Transcoding," in Simon Harper and Yeliz Yesilada, editors, Web Accessibility: A Foundation for Research, Human-Computer Interaction Series, Springer, 2008, pp. 231–260.

[2] A. Brown, C. Jay, and S. Harper, "Audio access to calendars," Proc. ACM 7th International Cross-Disciplinary Conference on Web Accessibility (W4A'10), ACM, Apr. 2010, doi:10.1145/1805986.1806028.

[3] O. Drümmer, "How feasible is text customization for PDF documents?," in Text Customization for Readability, W3C Online Symposium, Nov. 2012. Available from: http://www.w3.org/WAI/RD/2012/text-customization/t6 [retrieved: August, 2013].

[4] S. L. Henry, "CSS for Readability: Analysis of user style sheets to inform understanding users' text customization needs," in Text Customization for Readability, W3C Online Symposium, Nov. 2012. Available from: http://www.w3.org/WAI/RD/2012/text-customization/r14 [retrieved: August, 2013].

[5] G. Legge, "Psychophysics of Reading," Lawrence Erlbaum Associates, Mahwah, New Jersey, 2006.

[6] Mozilla Firefox Add-on Builder and SDK, 2013. Available from: https://addons.mozilla.org/en-US/developers/ [retrieved: August, 2013].

[7] S. Panchanathan, T. McDaniel, and V. N. Balasubramanian, "Person-Centered Accessible Technologies: Improved Usability and Adaptation through Inspirations from Disability Research," Proc. International Workshop on User Experience in e-Learning and Augmented Technologies in Education (UXeLATE'12), ACM, Oct. 2012, pp- 1-6, doi:10.1145/2390895.2390897.

[8] L. Rello and R. Baeza-Yates, "Optimal Colors to Improve Readability for People with Dyslexia," in Text Customization for Readability, W3C Online Symposium, Nov. 2012. Available from: http://www.w3.org/WAI/RD/2012/text-customization/r11 [retrieved: August, 2013].

[9] L. Rello, M. Pielot, M. Marcos, and R. Carlini, "Size matters (spacing not): 18 points for a dyslexic-friendly Wikipedia," Proc. ACM 10th International Cross-Disciplinary Conference on Web Accessibility (W4A'13), ACM, May 2013, doi:10.1145/2461121.2461125.

[10] P. Salomoni, S. Mirri, S. Ferretti, and M. Roccetti, "Profiling Learners with Special Needs for Custom E-Learning Experiences, a Closed Case?," Proc. ACM 4th International Cross-Disciplinary Conference on Web Accessibility (W4A'07), ACM Press, May 2007, pp. 84-92, doi:10.1145/1243441.1243462.

[11] P. Swapna Raj and B. Ravindran, "Personalized Web-page Rendering System," Proc. International Conference on Management of Data (COMAD 2008), Dec. 2008, pp. 30–39, doi:10.1.1.143.5393.

[12] M.A. Tinker, "The Legibility of Print," Iowa State University Press, 1963.

[13] Y. Yesilada, S. Harper, and S. Eraslan, "Experiential Transcoding: an eye tracking approach," Proc. ACM 10th International Cross-Disciplinary Conference on Web Accessibility (W4A'13), ACM Press, May 2013, doi:10.1145/2461121.2461134.

[14] W. M. Watanabe, A. Candido, M. A. Amâncio, M. de Oliveira, T. A. S. Pardo, R. P. M. Fortes, and S. M. Aluísio, "Adapting web content for low-literacy readers by using lexical elaboration and named entities labeling," Proc. ACM 7th International Cross-Disciplinary Conference on Web Accessibility (W4A'10), ACM Press, May 2010, doi:10.1145/1805986.1805998.

[15] C.J.C.H. Watkins, "Learning from Delayed Rewards," Ph.D. thesis, Cambridge University, 1989.

[16] K. Wecel, T. Kaczmarek, and A. Filipowska, "Scalable Adaptation of Web Applications to Users' Behavior," Proc. 4th International Conference on Computational Collective Intelligence: technologies and applications (ICCCI'12), Springer-Verlag Berlin, Heidelberg, Nov. 2012, pp. 79-88, doi:10.1007/978-3-642-34707-8_9.

[17] WURFL ScientiaMobile, WURFL, 2013. Available from: http://www.scientiamobile.com/ [retrieved: August 2013].