Exploiting Reinforcement Learning to Profile Users and Personalize Web Pages

Stefano Ferretti, Silvia Mirri, Catia Prandi, Paola Salomoni
Department of Computer Science and Engineering
Università di Bologna
Bologna, Italy
{s.ferretti, silvia.mirri, catia.prandi2, paola.salomoni}@unibo.it

***Abstract*—In this paper, we present a Web content adaptation system that is able to automatically adapt textual elements of Web pages, based on the user profile and preferences. The system employs Web intelligence to perform these automatic adaptations on single elements composing a Web page. In particular, a reinforcement learning algorithm, i.e. q-learning, based on the idea of reward/punishment is utilized as the machine learning system that manages the user profile. Based on it, the user profile is updated, so that automatic adaptations can be effectively performed while surfing the Web. We created a simulation scenario to test our approach over different users with specific preferences and/or different kinds of disabilities. Simulation results confirm the viability of the proposal.**

Keywords—content adaptation; Web personalization; user profiling; legibility; reinforcement learning

I. INTRODUCTION

Web content adaptation can provide great benefit to users. It has been observed that different kinds of personalization and (self-)adaptation of Web contents can be performed, which allow to tailor the way contents are presented to users, including their shapes and formats [1]. Usually, the typical approach, employed to adapt the layout of a Web page and the formats and shapes of contents within the page, consists in applying such transformations to all the elements which compose the entire Web page.

Actually, it is evident that a more user-centered approach should be exploited, in order to customize only the shape of some specific Web page elements, according to users' needs. Such an approach could have a strong impact, in particular for those users with some reading-related disabilities (i.e. people with dyslexia, users with low vision, users with color blindness, elderly people, etc.). Furthermore, adapting just some specific Web elements, instead of the whole page, means that it would be possible to transform only those parts of a Web content that represent an effective barrier to users. This way, the whole content and its layout are not distorted, and customization is just limited to those elements that really affect users' reading. Finally, this approach can provide benefits even to those users who are equipped with devices with different capabilities, such as tablets, smart-phones, and smart TVs. In this context, both readability and legibility are affected by different issues, such as Web text characteristics [2, 3, 4] and users' abilities [5, 6, 7].

The main aim of this work is to improve Web content legibility by adapting some text formatting characteristics (i.e., font size, font face, luminance contrasts, and so on) according to users' preferences and needs. To reach this goal we have designed a system, called ExTraS (EXperiential TRAnscoding System [8, 9]) which lets users adapt Web documents, tracking users' behavior, so as to learn and model their preferences and to automatically provide the best adaptation, tailored for each user, predicting his/her needs.

We have built a prototype of our system as a browser extension, as described in [9]. We have considered the adaptations of characteristics that help users in improving textual content legibility. They strongly depend on the specific reading difficulties of the user; in fact, an adaptation which may result effective for a specific user (with certain preferences) could be counterproductive for another one (with different preferences). With the aim of doing a strong personalization on the basis of user's needs, we decided to let the users adapt those Web page elements which represent an actual barrier for them, instead of automatically adapting the entire Web page. This avoids significant layout distortion, and, at the same time, it gives emphasis on those elements that the user selects. Automatic adaptations are performed on the basis of a user profile that keeps track of the his/her needs, taking into account both the user's experiences and behaviors.

In order to understand user's experience and to learn user's preferences, we have exploited a machine learning mechanism, the Reinforcement Learning one, based on the idea of reward/punishment. In particular, we have selected the Q-learning algorithm [10], which focuses on developing a specific action in a specific state. According to this technique, the agents keep track of the experienced state-action pairs by managing a table (called Q-table) which represents the experience data as behavioral rules [10]. Thanks to it, 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 (including those ones that represent a barrier for him/her). Thus, the Web content adaptation will be more user oriented, better meeting user's needs.

To test our system, we have integrated a simulator and a framework which applies Q-Learning [11], as we have tailored to our context. This allowed us to evaluate our system thanks to a simulation campaign. In this paper, we present the simulation scenarios, we describe the evaluations we have conducted, and

we report the results we have obtained, that confirm the viability of our proposal.

The remainder of the paper is organized as follows. Section II describes some main related work; Section III presents the system architecture and some main issues of the prototype we have developed; the users profiling management and the learning system are described in Section IV. Section V presents the simulations we have conducted to evaluate how our system learns users' preferences and the related results. Finally, Section VI concludes the paper presenting further work.

II. Background and Related Work

Our work considers several issues, which are briefly described in this section: improving Web pages legibility with font adaptation; adaptation and personalization of digital and Web contents and services; the use of machine learning techniques and algorithms to track and understand users' behavior.

According to the definitions in the literature, legibility is related to perceiving text by distinguishing each character from all other ones in the font, without any ambiguity. Hence the more text letters are distinguishable, the more such a text is legible [3, 4]. A different issue is the readability, which is related to reading and understanding textual information: the more a text is complex (with difficult words, long paragraphs, acronyms, abbreviation, technical terms, foreign sentences, etc.) and the less a text is readable [3, 4]. Summing up, legibility refers to text perception, while readability refers to its comprehension. In this context, the reading activity can be strongly affected by textual characteristics, such as font face, text size, background and foreground colors, alignments, paragraphs, words and letters spacing [2, 5]. Online reading abilities and textual characteristics, which better support users in such a kind of activities, are the topics of several studies, taking into account also the device in use (including mobile [12] and smart devices [13]), as well as specific users contexts or condition (including reading disabilities [6], visual impairments [7] and ageing [5]). Outcomes from these previous studies have been taken into account in our work.

The personalization and adaptation of digital Web contents and services are at the basis of several studies. The need for personalization ranges from the need for the use of different formats and aspects (due to different devices, context of use, or specific preferences and conditions [14, 15]) to the need for different content (recommending resources that can be of interest for the users, according to the preferences and the behaviors they have already shown somehow). Several works have been done in this field. The personalization of Web pages with the aim of better exploiting them by means of mobile devices is the goal of [16]. Another work in this direction, where users' preferences are taken into account, is [17]. User-centric personalization of Web sites, by means of personalized content recommendations is at the basis of [18]. In works like this latter one, user profiling is a fundamental activity, because it can drives the personalization and adaptation of content and services [19]. Users' profile and model can also be built according to their behaviors.

Users' behaviors as personalization-drivers have been exploited in [20]; in this case, the author proposes an approach to construct a dynamic user model, strongly based on user's behavior, which drives the personalization of media augmentation, accessed by using some mobile devices. Analogously to these cited solutions, our work exploits the idea of adapting Web contents on the basis of users' experience. In particular, we track past users' adaptations so as to learn users' preferences and to provide (and/or suggest) adequate personalization. This approach has been inspired by recommendation systems used in e-commerce Web sites, social networks, search engines, etc. A similar approach has been used in [21], where the authors exploited a reinforcement learning algorithm in order to customize the rendering of advertisements in Web pages, on the basis of users' preferences. Learning and predicting users' preferences, so as to drive recommendation systems is also the main aim of [22]. The reward/punishment approach has been exploited also in [23], where the authors propose the use of the Q-learning algorithm for modeling the behavior of agents in simulations.

III. System Architecture

Our system architecture is structured as shown in Figure 1. A prototype has been implemented as a Firefox extension by means of the Mozilla SDK, which is responsible to adapt HTML documents. The software architecture is structured in three modules: the Profiling Module, the Learning Module and the Adaptation Module.

The Adaptation Module is in charge of transcoding HTML tags, attributes and their related values and of personalizing CSS rules. To adapt contents based on the user profile, the system is capable to inject new tags or attributes into the content, and it is able as well to substitute original tags or attribute values with customized ones.

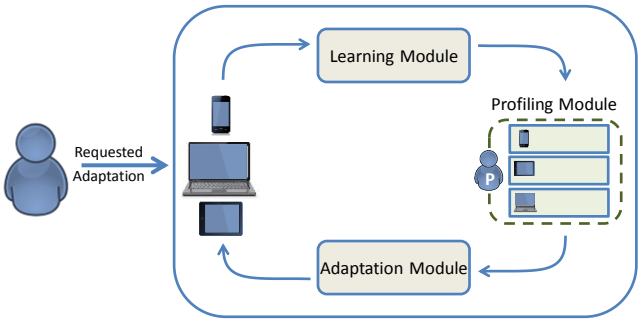

Fig. 1 - System Architecture

Users can exploit a contextual menu to set adaptations on an HTML page. While the adaptation process transcodes the HTML and/or the CSS code of the page, the behavior of the user is tracked, with the aim of learning his/her preferences (Learning Module in the figure). All the user preferences are stored in a profile, managed by the Profiling Module shown in the figure. This allows to apply or propose suitable adaptations automatically during future interactions. The profile created automatically, while surfing the Web, is locally stored on the

device. If a user has different devices, and the profile is shared among these devices, updates are synchronized, as in typical cloud-based approaches.

The user profile is structured as an XML document. It is composed of different parts, related to the devices the user exploits. In each of these parts, the system stores text characteristics (as tags) the user has asked for adaptations or the user has discarded (respectively user's preferences or barriers). Thus, each element corresponding to a text characteristics has associated a value (hereinafter referred as the "v" attribute) that identifies it, and a number (hereinafter referred as the "r" attribute). Such an "r" number tracks the reward/punishment related to the adaptation of the associated "v" element, asked by the user. Hence, such "r" value varies based on the user behavior.

Currently, the prototype has been tested on laptops (equipped with different operating systems) and on Samsung Galaxy Tab 2 devices, equipped with Android 4.0.4 and Firefox 27.0. More details about our prototype, some examples of adaptation and some screenshots of the system in action are reported in [8] and [9].

The focus of this work is on the learning process that drives the creation and automatic modification of the user profile. Its functioning is described in the following section.

IV. Learning System and Profile Management

A reinforcement learning algorithm, based on the idea of reward/punishment is utilized as the machine learning system that manages the user profile [8]. Based on it, the user profile is updated, either by adding new preferences/barriers, or by updating reward/punishment values associated to existing elements of the profile.

We already mentioned that the user profile is composed of different elements that describe the preferences of the user on the use of some type of text, e.g., font family, font size, colors, etc. To describe this, each element has a "v" value that identifies the specific textual characteristics (i.e. "Arial" for font family, "14" for font size, etc.), and an "r" value that tracks the preferences of the user with respect to "v". In the following, when we say that the "user has discarded an element v", we mean that the user decided to not visualize some content using that textual characteristics "v", considering them a barrier, which affects the content legibility. Conversely, we will state that "the user has chosen" to exploit the "v" element, meaning that he/she accepted to visualize some content that have that specific textual characteristics. The "r" attribute is related to the reward/punishment assigned by our system and can assume different values; in essence, we sum up three possible cases:

1. $r < 0$: this means that the user has discarded the "v" value for a given element;
2. $r > 0$: this means that the user has chosen the "v" value for a given element;
3. $r = 0$: this means that the "v" value for a given element has been chosen and discarded the same amount of times.

Thus, the idea is that preferences of the user are identified by the system based on discarded and chosen text formatting characteristics, tracking even those elements that represent a barrier for that user. The user profile is initially empty; the system learns preferences and updates the profile while the user surfs the Web. In particular, the requirement of adaptations and changes to some elements in a Web page corresponds to punishments to the values for these elements, since the user considers them as barriers. A more detailed description (as well as an example of a user profile) can be found in [9]. Further, our system will be able to perform automatic adaptations; when these proposed adaptations are accepted by the user, the values related to the adapted elements obtain a reward. These punishments and rewards are managed based on reinforcement learning.

The specific algorithm in charge of managing these punishments/rewards is Q-learning [10]. In essence, this scheme allows to learn the optimal policy to accomplish, based on the history of interactions of the system with the environment. The history is a sequence of state-action-rewards $<s_0,a_0,r_1, s_1,a_1,r_2, ...>$, meaning that when the system is in a state s_i, it takes the action a_i, obtaining a reward r_{i+1}. For each of these actions, the algorithm updates an estimation $Q(a,s)$ of the reward obtained by taking the action a when the system is in state s. Each time the system takes an action, given a state, and it receives a reward, such an estimation $Q(a,s)$ is updated based on the following equation:

$$Q[s,a] \leftarrow Q[s,a] + \alpha(r + \gamma \max_{a'} Q[s',a'] - Q[s,a]) \quad (1)$$

where α is a step rate; r is the observed reward; s' is the state where the system goes by taking the action; $\gamma < 1$ is a parameter, that works as a discount value, which serves to give a weight to the estimation of the maximum reward $\max_{a'} Q[s',a']$ that system can measure by taking some future action in the novel state s', based on its current information.

The general algorithm executed by our system is described in the following.

1) When a user opens up some few Web pages, then he/she can adapt some characteristics. At this initial stage, the system just tracks his/her behavior and starts to learn his/her preferences. Moreover, it starts to assign reward to the chosen characteristics, according to the related state, so as to start building the user's profile.

2) When a user opens up a new Web page, then the system parses the DOM and the related style rules, taking into account the user's profile.

3) All characteristics of elements in the DOM are matched with the user profile. In particular, all those characteristics that the user has discarded (with a low "r" value in the profile) are collected.

4) For each characteristic, the system maintains a negative threshold value "t" that is exploited as follows. For each element with a low "r" value, the adaptation that has associated the highest reward is considered. In particular:

a) if $t < r < 0$: the system proposes such adaptation to the user; if the user accepts it, then a reward is assigned; else, a punishment is associated;

b) otherwise ($r < t$), the system automatically adapts such characteristics, by substituting them with the ones the user prefers. If the user ignores the automatic adaptation (i.e. it is accepted), then a reward is assigned. Else, the user rejects the adaptation; thus a punishment is associated.

5) The user can in turn adapt some element by himself/herself. In this case, the system observes the user's behavior and assigns related reward/punishment. Unchanged characteristics receive no reward.

6) All the updated rewards and punishments are stored into the user profile.

The system repeats phases from 2 to 6 any time the user opens up a Web page.

V. EVALUATION

In order to assess the viability of our proposal, we have developed a simulator which mimics the user behavior and we have integrated it with a framework that implements the reinforcement learning algorithm. In SubSection A we describe the simulations we have performed and in SubSection B we present the related results.

A. Simulation

A simulation has been performed in order to obtain a quantitative evaluation of the benefits provided by the intelligent adaptation system. The simulator is in charge of mimicking user behaviors while surfing the Web. Users might have some kinds of disabilities, since this corresponds to some specific preferences (and in many cases, technological barriers) to certain types of characteristics of styles, fonts and dimensions, associated to elements composing some Web pages. In particular, we have modeled three different kinds of users: users with dyslexia; users with low-vision; elderly users. This choice has been driven by the fact that such kinds of users have different and very specific needs in terms of reading activities, and then we decided to test our system in these challenging contexts. Of course, the simulator can be extended to other contexts.

Each user has its own characteristics, grouped as a profile, corresponding to the most typical preferences that users might want when accessing a given content, avoiding those text characteristics which can represent a barrier, affecting legibility for them, according to the literature [5, 6, 7]. Hence, we have created three different user models and during the simulation we have considered the following preferences for such models, to be associated to the text elements of the Web pages.

We assume that a user with dyslexia commonly prefers to read text with an Arial font, or a San Serif font family. He/she prefers that text elements are displayed with a font size equal or higher than 18 points. As concerns the text alignment, a left alignment is preferred while a justified alignment is highly deprecated [6].

A user with low vision, instead, benefits from San Serif type fonts; font size should be around 16-18 points. A left alignment is preferred, with a line height set to 1.5 [7].

Finally, a 14-point font size has a significant improvement in legibility for elderly people; they also prefer san serif fonts, in particular Arial and Verdana (as a second choice) [5]; moreover, left alignment is preferred, as well as space between lines, paragraphs and around clickable targets (such as links and buttons, so that each one is easy to target and hit separately) [3].

In order to assess whether (and how) the system is able to react to changes in user preferences, we also implemented a sort of inconsistent user, that from time to time gives some feedbacks to our system which are not compliant to his/her user model. Finally, we have also implemented a user who changes his/her preferences during the simulation. As an example, this can be the case of an elderly user.

The simulator mimics a user behavior, browsing the Web among a set of the 10 most visited pages, taken from the Alexa top Web sites list, according to ranking data they have gathered in February 2014 [24]. The set of Web pages was: Google home page, Google search page (http://www.google.com), Facebook news, Facebook user's wall (http://www.facebook.com), YouTube home page, YouTube video page (http://www.youtube.com), Yahoo! home page, Yahoo! result page (http://www.yahoo.com), Wikipedia home page, Wikipedia item page (http://wikipedia.org).

To implement the reinforcement algorithm, exploited by the Learning Module, the Piqle framework was utilized [11]. In order to do that, we have customized Piqle, so as to simulate our system and the users' behavior. Summing up, we have mainly implemented the following issues:

- Three main entities simulate the adaptation process: Environment, State and Action. The user directly interacts with the Environment. A State is the collection of the text characteristics of the Web page the user is surfing, while an Action is an adaptation of a specific text characteristic for a specific element of the Web page.
- The reward/punishment assigned to the *(State, Action)* performed is not generated by the Environment (as usual), but it is computed according to the user's feedback (on the basis of his/her model).
- The system starts to learn after an initial observation phase, during which it just observes and tracks user's behaviors, rewarding to the chosen characteristics (according to the related state) and understanding the barriers that affect user's reading. It starts to propose and automatically perform adaptation only after this phase.
- A sort of punishment is assigned also to the discarded text characteristics (identified as barriers), so as to enforce the need of adapting them.

B. *Results*

In order to evaluate how our system learns users' preferences, we have conducted several simulations for each of the three user models we have described in Section V.A. During such simulations, the users can enjoy from 1,000 to 500,000 Web pages. After such a simulation campaign, we are able to conclude that the system learns the user's preferences, independently from the simulated user.

For each simulation, we have counted the adaptations the system has performed that were not compliant to the user profile. This way, we can observe the number of errors the system does, while it learns. Such a value represents the distance between the simulated user profile and the user model the system learns during the trials. Simulation steps are represented by browsed Web pages. When the errors are 0, then the system has learnt the users' preferences. We have observed that the average number of browsed Web pages after which the number of errors (in terms of adaptations the user rejected, because they are not compliant to his/her profile) tends to zero changes according to the initial phase of observation. This means that the longer is the observation phase and the less the number of wrong adaptations the system performs.

Figure 2 shows a single simulation of a user with dyslexia, which browses 1,000 Web pages. The user does not change his/her preferences during the trial and, in this case, our system proposes 10 errors after 14 browsed Web pages, after that the number of errors tends to zero.

In order to better test our system, we have simulated even users who change their minds during the trials. In particular, we have simulated some users with a certain probability (expressed in percentage, from 0.1% to 20%) of giving feedbacks which are not compliant to their profile. Figure 3 shows a single simulation of a user with low vision who gives a 1% of inconsistent feedback, in a trial with 1,000 Web pages. The peaks in the plot correspond to the situations when the user does not provide feedbacks compliant to his/her profile. In these cases, the system applies some sort of wrong adaptations contextually, due to those inconsistent user's behaviors. Such inconsistent behaviors could be due to some contextual situations that can affect user's reading, such as, for instance, a different ambient light. After those peaks, the system continues to perform adaptations compliant to the profile and then the user accepts them as usual, with no errors. No additional learning phase is required, since the profile has not been changed.

Moreover, we have simulated users who change their minds just once during the trials. Figure 4 show a single simulation of an elderly user who asks for bigger font size dimensions after 500 pages. The peak shown in the plot corresponds to that change of preferences. After that, an additional learning phase is needed. This means that the system applies some adaptations which are still compliant to the old preferences, but not to the new ones. Hence, the user has to reject such wrong adaptations, so as to let the system learn his/her new different needs. In this case, since only one preference has been changed, then the number of steps needed, so that the number of errors results negligible, is less than the initial ones. The same is for the number of errors the system does at the very first step the user changes his/her mind about font size dimensions. In cases like this one, we have observed that the later the change of preferences happens and the higher is the number of errors that the system commits after this modification in the user's profile.

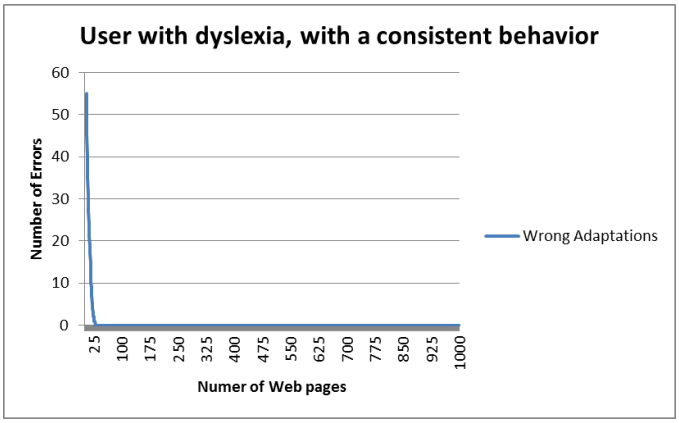

Fig. 2 – User with dyslexia, with a consistent behavior

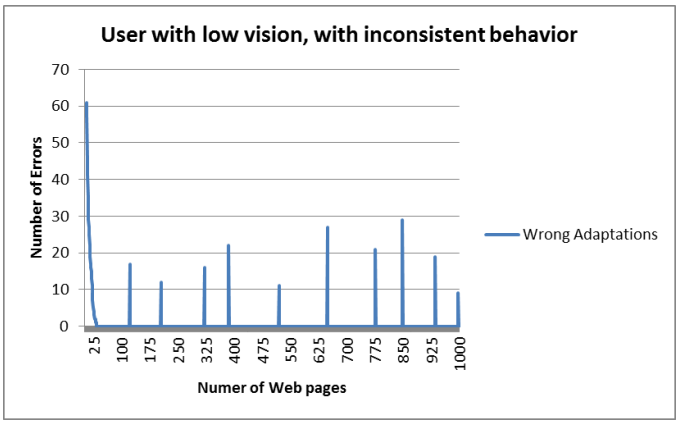

Fig. 3 – User with low vision, with inconsistent behavior

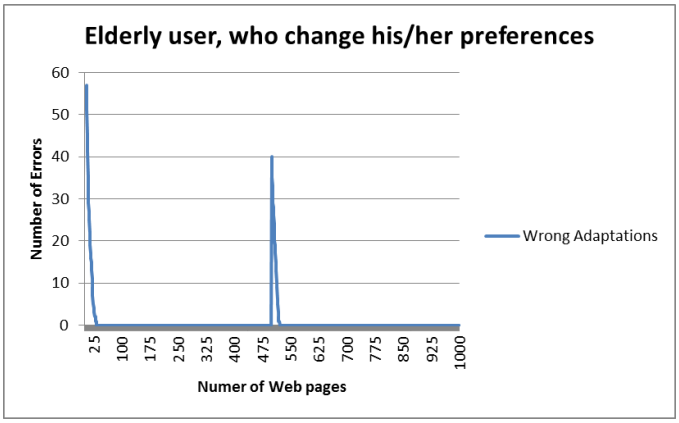

Fig. 4 – Elderly user, who changes his/her preferences

VI. Conclusion and Future Work

In this paper, we have presented a system based on reinforcement learning, with the aim of providing personalization of Web pages, according to users' preferences and needs. The goal of our system is to improve Web pages legibility, by adapting only those textual elements that represent a barrier to users reading. Thanks to the Q-learning algorithm, our system tracks users' behavior and then applies the best adaptations, tailored for each specific user.

We have tested our system by integrating a simulator and a version of the PIQLE framework we have customized according to our context. Results from the simulation are presented in this paper, which confirm the viability of the proposal.

Further work is needed, that might exploit a more complex user profile, describing barriers and preferences for more textual characteristics. This would increases the Q-table dimensions, providing more complexity. In order to face it, we will combine Q-learning with other techniques, such as Decision Tree learning. Another interesting future work consists in comparing the techniques we have employed to generate and manage the user profile with other methods (such as the Top-N recommendation), taking into account that the user preferences can change over time. With the aim of improving personalization and learning activities, we will exploit techniques for sharing user profiles, computing their similarities and proposing adaptations to users of the basis of similar profiles (i.e. collaboration filtering techniques).

Finally, an evaluation with real users might allow to better understand the real appreciation of users to our system and to test the amount of activities they are ready to perform before the system learns their preferences. We are now planning a campaign with users with reading related disabilities and users equipped with different mobile devices.

Acknowledgment

The authors wish to thank Alberto Fariselli for his precious support.

References

[1] 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.

[2] D. Beymer, D. M. Russell, and P. Z. Orton, "An eye tracking study of how font size, font type, and pictures influence online reading," Proc. 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction (BCS-HCI'08), September 2008, pp. 15-18.

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

[4] M.A. Tinker, "The legibility of print," Iowa State University Press, 1963.

[5] M. Bernard, C.H. Liao, and M. Mills, "The effects of font type and size on the legibility and reading time of online text by older adults," Proc. ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI2001), ACM Press, March-April 2001, pp. 175-176.

[6] L. Rello, G. Kanvinde, and R. Baeza-Yates, "Layout guidelines for web text and a web Service to improve accessibility for dyslexics," Proc. 9th ACM International Cross-Disciplinary Conference on Web Accessibility (W4A'12), ACM Press, April 2012.

[7] E. Russell-Minda, J.W. Jutai, J. Graham Strong, Kent A. Campbell, Deborah Gold, Lisa Pretty, and Lesley Wilmot, "The legibility of typefaces for readers with low vision: a research review," Journal of Visual Impairment & Blindness, Volume 101, Number 7, July 2007, pp. 402-415.

[8] S. Ferretti, S. Mirri, C. Prandi, and P. Salomoni, "User centered and context dependent personalization through experiential transcoding," Proc. IEEE Consumer Communications and Networking (CCNC 2014), Workshop on Networking Issues in Multimedia Entertainment (NIME'14), IEEE Computer Society, January 2014.

[9] S. Mirri, C. Prandi, and P. Salomoni, "Experiential adaptation to provide user-centered web content personalization," Proc. IARIA Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC2013), October 2013, pp. 31-36.

[10] C.J.C.H. Watkins and P. Dayan, "Q-learning", Machine Learning, Vol. 8, no. 3, pp. 279-292, 1992.

[11] Piqle, "Platform for implementing Q-Learning experiments", http://piqle.sourceforge.net/, [last retrieved: March 2014].

[12] B. Schildbach and E. Rukzio, "Investigating selection and reading performance on a mobile phone while walking," Proc. 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI'10), ACM Press, September 2010, pp. 93-102.

[13] P. Biswas and P. Langdon, "Investigating the accessibility of program selection menus of a digital TV interface," Proc. International Conference on Human Computer Interaction (HCII2011), July 2011, pp. 425-434.

[14] K. Z. Gajos, D. S. Weld, and J. O. Wobbrock, "Automatically generating personalized user interfaces with supple," in Artificial Intelligence, vol. 174, no. 1213, pp. 910 – 950, 2010.

[15] D. Wolff, M. Schaaf, S. Grivas, and U. Leimstoll, "Context-aware website personalization," in Knowlege-Based and Intelligent Information and Engineering Systems, ser. Lecture Notes in Computer Science.Springer Berlin Heidelberg, vol. 6882, 2011, pp. 51–62.

[16] C. Wei, H. Lee, L. Molnar, M. Herold, R. Ramnath, and J. Ramanathan, "Assisted human-in-the-loop adaptation of Web pages for mobile devices," Proc. IEEE Signature Conference on Computers, Software, and Applications (COMPSAC 2013), IEEE Computer Society, July 2013, pp. 118-123.

[17] G. R. Santhanam, S. Basu, and V. Honavar, "Preference based service adaptation using service substitution," Proc. IEEE Web Intelligence (WI 2013), IEEE Computer Society, November 2013, pp. 487-493.

[18] K. Koidl, O. Conlan, and V. Wade, "Towards cross site personalisation," Proc. IEEE Web Intelligence (WI 2013), IEEE Computer Society, November 2013, pp. 542-548.

[19] O. Bursa, Ö. Can, and M. O. Ünalir, "User Profiling for Policy Management in Social Communities," Proc. IEEE Signature Conference on Computers, Software, and Applications (COMPSAC 2012), IEEE Computer Society, July 2012, pp. 348-349.

[20] S. Loeb, "Fluid user models for personalized mobile apps," Proc. IEEE Consumer Communications and Networking Conference (CCNC 2013), IEEE Computer Society, January 2013, pp. 705-708.

[21] P. Swapna Raj and B. Ravindran, "Personalized web-page rendering system," Proc. International Conference on Management of Data (COMAD 2008), December 2008, pp. 30-39.

[22] Y. Ren, G. Li, and W. Zhou, "Learning user preference patterns for Top-N recommendations," Proc. IEEE Web Intelligence (WI 2012), IEEE Computer Society, December 2012, pp. 137-144.

[23] R. Junges and F. Klügl, "Programming agent behavior by learning in simulation models," in Applied Artificial Intelligence, Vol. 26, no. 4, pp. 349-375, 2012.

[24] Alexa Internet, Alexa Top 500 Global Sites. Alexa Internet. http://www.alexa.com/topsites [last retrieved: March 2014].