



PUC

ISSN 0103-9741

Monografias em Ciência da Computação
nº 30/09

Collaborative Web Scripting for Improved Accessibility

**Chantal Intrator
Clarisse Sieckenius de Souza**

Departamento de Informática

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900
RIO DE JANEIRO - BRASIL**

Collaborative Web Scripting for Improved Accessibility^{*}

Chantal Intrator and Clarisse Sieckenius de Souza

{cintrator, clarisse}@inf.puc-rio.br

Abstract. This paper presents WNH, the Web Navigation Helper, which is based on CoScripter, a collaborative macro recorder for the web. WNH helps blind and functionally illiterate users interact with websites by interpreting previously generated scripts for achieving various kinds of tasks. We report the results of a preliminary empirical study carried out during formative evaluation steps and discuss some of the challenges and promises associated to our findings.

Keywords: Web navigation agents for users with special needs; Accessibility for blind users; Accessibility for Functionally Illiterate Users; Web Macros; CoScripter.

Resumo. Este artigo apresenta o WNH (Web Navigation Helper), que é baseado no CoScripter, um gravador colaborativo de macros para a Web. O WNH auxilia usuários com deficiências de visão e alfabetismo funcional a interagirem com websites, interpretando scripts previamente gerados para realizar vários tipos de tarefas. Reportamos aqui os resultados de um estudo empírico preliminar, realizado durante as etapas de avaliação formativa. Discutimos também alguns dos desafios e premissas associadas a nossos achados.

Palavras-chave: Agentes de navegação na Web para usuários com necessidades especiais; Acessibilidade para Cegos; Acessibilidade para Analfabetos Funcionais; Macros para a Web; CoScripter.

* Este trabalho resulta de um *Joint Study Agreement* selado entre o SERG e a IBM Research de Almaden, California. Clarisse de Souza recebeu bolsa de pesquisador visitante da IBM Research Almaden (Ago2007) e tem apoio de pesquisa do CNPq. Chantal Intrator recebeu bolsa de mestrado CAPES Modalidade II. O trabalho contou também com a valiosa colaboração de Ana Pavani, Departamento de Elétrica da PUC-Rio, que cedeu o software JAWS para os experimentos realizados. Além disto, agradecemos a todos os participantes voluntários, sem os quais este estudo não teria sido possível.

In charge of publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: <http://bib-di.inf.puc-rio.br/techreports/>

TABLE OF CONTENTS

1 INTRODUCTION	1
2 RELATED WORK.....	1
3 COSCRIPTER.....	2
4 THE WEB NAVIGATION HELPER.....	3
5 EMPIRICAL EVALUATION.....	5
6 CONCLUSIONS AND FUTURE WORK.....	7
REFERENCES	8

1 INTRODUCTION

Web accessibility for users with special needs has been drawing the attention of a growing community of researchers and professionals. The W3C Web Accessibility Initiative has developed guidelines for web site design [15] and user agent accessibility [18]. They consolidate a considerable volume of research results on accessibility and universal usability. Although these two topics usually make one think first of how to attend to people with sensory disabilities, Zajicek and Edwards [19] remind us that universal usability involves removing barriers in many fronts other than physical capacities, including: economics, education, literacy, politics, and culture.

This paper presents the Web Navigation Helper (WNH), a user accessibility agent prototype based on CoScripter [10] [13]. WNH operates with *coscripts*, web automation scripts collaboratively generated by users without disabilities and made available in a special repository [10]. It helps blind and functionally illiterate users interact with websites by interpreting previously generated scripts for achieving various kinds of tasks, and intermediating user-website interaction. WNH has different interface features for the two communities of users, which we evaluated empirically with a small group of users.

In the next sections we briefly discuss related work (section 2). Then we introduce CoScripter (Section 3) and WNH (section 4), before we discuss our evaluation methods, procedures and results (section 5). The last section of the paper (section 6) presents our concluding remarks and future work plans.

2 RELATED WORK

In Vanderheiden's opinion [16], we need to explore cooperative and collaborative approaches to accessibility in order to narrow the gap between the ever-growing volume of new information and services on the web and the number of people that cannot access them, on the other. One way to do it is, as he proposes, to develop a common open-source technical core for assistive technologies. The other is to engage communities of volunteers in the effort of making the web more *navigable* for users with special needs. Bigham and Ladner [3], for example, propose that web scripting tools could be used to generate more accessible versions of web pages, especially for the blind. The same approach was used in HearSay3 [5], where collaborative web page labeling is used to accelerate navigation to targeted links. CoScripter, a collaborative macro recorder for the web [10] was not specifically designed to improve accessibility, but by automating web navigation for all users, it can also help users with special needs to accomplish various kinds of tasks, using scripts produced by other users [11].

Other approaches to improving web accessibility propose to use agents and various types of machine learning and artificial intelligence techniques. Pontelli and Son [14] have built an agent to help visually-impaired users with *e-commerce* applications. They use conceptual structures to represent navigational semantics, planning and problem-solving techniques to help these users avoid visiting all points of the XPath with screen readers. Harper and Patel [9] use gist summaries of web pages generated 'on the fly' to help visually-impaired users have accelerated access to relevant information. The current version of HearSay [15] uses machine learning techniques to support context-directed browsing for screen-reader users. TrailBlazer [4], which also extends CoScripter for accessibility purposes, uses programming by demonstration techniques to suggest navigation moves to blind users.

There is of course a vast volume of work on accessibility, especially for blind users. ACM Transactions on Accessible Computing Volume 1(2), from October 2008, includes five commentary papers discussing progress and lags in accessibility in the last two decades. Here we only refer to work directly related to the focus of this paper. So, we should highlight Walton and Vukovic's work on cultures, literacy and the web [19]. The authors remark that "Web use is situated within distinct cultures of reading and writing", and that Web designers and developers tend to assume levels of literacy that are absent in very large portions of the world population. In Brazil, for example, it is estimated that nearly 30% of Brazilians from 15 to 64 years of age can only read and write very simple text (short sentences, direct syntax, most information explicitly expressed, using little or no subordinate clauses, etc.).

Although people with disabilities have specific cognitive skills and needs (see [2], for example), the very idea of universal usability [20] has somehow encouraged researchers to look for new designs, techniques and tools that have the potential to benefit more than one community of users with special needs. For instance, Evett and Brown [8] studied text format and style recommendations of the Royal National Institute for the Blind and the British Dyslexia Association, and concluded that textual web content written according to certain guidelines should benefit not only blind and dyslexic users, but also Web users in general. In this context, research on text simplification and summarization has become attractive to accessibility research (see [9], [6] and, for Brazilian Portuguese, [1]).

Our work also aims to benefit more than one community of users with a single technology. However, instead of working with text and information, we are exploring interaction with web pages.

3 COSCRIPTER

CoScripter is a macro-recorder for the web, implemented in JavaScript and XUL as an extension for the Mozilla Firefox browser [13]. It allows end-user automation of procedures through recording and scripting, and stores scripts on a shared central Wiki [10]. Once recorded, scripts can be easily reproduced. Filling in forms, clicking on buttons, navigating to specific URLs are some of the actions that can be easily automated.

The CoScripter interface is a sidebar located on the left side of the browser. Every script is designated by its owner as private or public. Public scripts can be seen and used by all members of the CoScripter community, while private ones are accessed and viewed only by their creators. When visiting a site, CoScripter exhibits to users a list containing all public scripts related to that site. Should users wish to run a script, they must just click on its name to load the script in CoScripter's sidebar, and click on "Run" to execute it.

In Figure 1 we show a public *CoScript* associated to the ACM Digital Library URL. The sidebar is divided into three segments: interface controls (top left), script area (center left, showing the command line that CoScripter is ready to execute), and the personal database area (bottom left, where private login and password information is stored). The scripting language syntax is very similar to natural English (in Figure 1 we see commands like 'click the "Login" link' and 'enter your ACM username into the "Web Account:" textbox'). The use of pronouns like 'you' and 'your' is an important directive for CoScripter. 'You' is interpreted as 'do nothing and wait for the user to interact with the page and tell he/she is done by hitting the "Run" button'. 'Your <expression>' is interpreted as 'pick up the value of variable named "<expression>" in the

user's personal database'. In the script shown in Figure 1, 'your ACM username' is replaced with "little me" and 'your ACM password' is replaced with "guess_what".

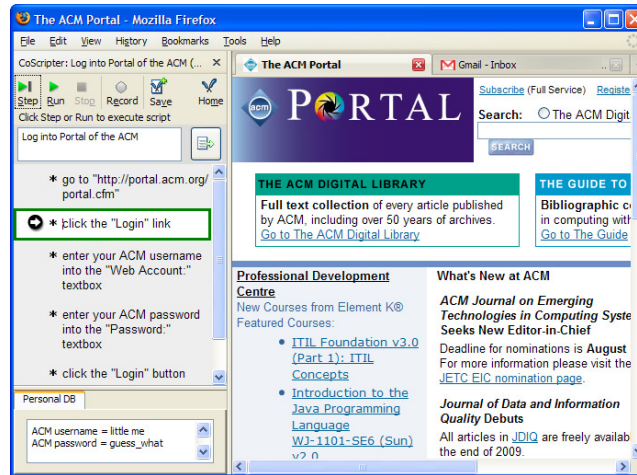


Figure 1. The CoScripter sidebar interface.

The most obvious advantages of CoScripter are faster navigation (because of automation) and sharing of *how-to* knowledge with a community of users (script commands are actually instructions for interacting with the website). However, from a technological perspective, CoScripter is an interaction *mediator*, whose dialog with the user can be considerably customized, as we will show in the next section.

4 THE WEB NAVIGATION HELPER

WNH is a user agent built on top of CoScripter. It provides specialized mediation for blind (WNH-see) and functionally-illiterate users (WNH-read). Both mediators can be seen as wrappers around CoScripter, adding some new features and filtering out others that may distract the targeted end users.

In Figure 2, we see a screen shot of WNH-see, with super-imposed arrows to facilitate our explanation. Of course the visual interface is completely irrelevant for the blind. The purpose of showing the screen is to call a sighted observer's attention to what non-sighted users are listening to with screen readers. The dialog box (see where the longer arrow is pointing) tells the user that a new step of the process has just been executed (step 3 of 14) and asks whether the user wants to proceed, ask for help, or cancel execution. The highlighted element of the original page (see where the shorted arrow is pointing) is produced by CoScripter. It shows the focus of interaction executed by the current script command. The script commands are not shown to the user; they can be accessed through 'help' dialogs if desired.

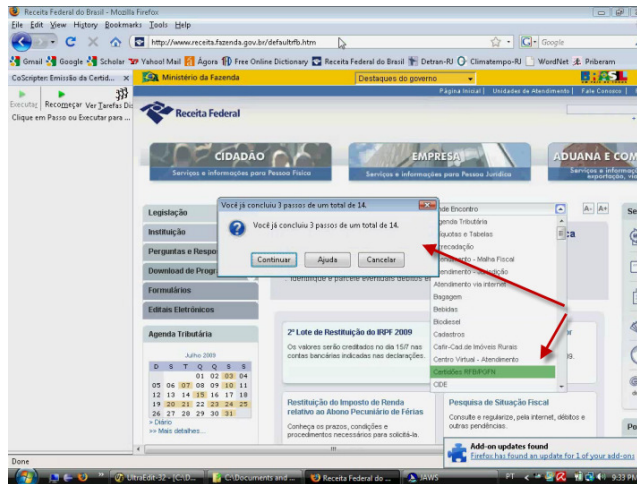


Figure 2. WNH-see helps a blind user jump through task-related links on web page and informs about progress.

In Figure 3, we see a screen shot of WNH-read. The visual interface is now relevant. We see that, as with WNH-see, most elements of the original CoScripiter sidebar have been taken out of the interface. Only three controls are presented to the user: Run (first on top left), start over (second) and see tasks you can do on this page (third). The mediator dialog is now richer, and contains more controls. In Figure 3, WNH-read is helping a functionally-illiterate user go past the CAPTCHA verification step. The mediator is asking the user to type in the numbers and letters he/she sees in the image on the page. The user can type them and proceed with 'OK', or cancel execution. Because CoScripiter is running on the background, the user may jump to the actual page and type in the numbers and letters on the CAPTCHA textbox itself (and not in the open dialog textbox). Although we haven't yet explored the switching back and forth between mediated and non-mediated conversation, the infrastructure for more flexible assisted/non-assisted conversation is there.

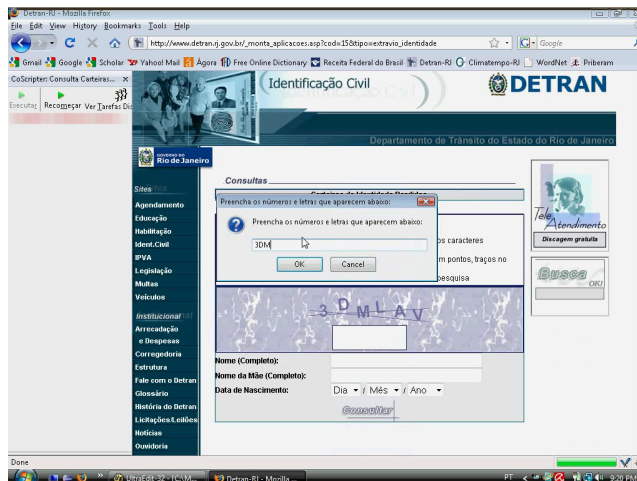


Figure 3. WNH-read helps a functionally-illiterate user through interaction with CAPTCHA.

Dialogs appearing in WNH-see and WNH-read are based on annotated *coscripts* produced by volunteers that do not have sight and literacy problems. Annotations follow mark-up patterns that specify how the mediator interface should look like. As of now, the patterns used for mark-ups are very rudimentary (only combinations of special characters used as prefix to CoScripiter's instructions). So, volunteers actually use a

third module of WNH, called WNH-support. In addition to supporting volunteer helpers as they record or write scripts for various pages and prepare the mediator dialogs to guide the targeted users through interaction, WNH support also provides a special tool (a web crawler) that periodically visits the pages to which scripts are associated and report changes that might have happened. These changes can cause the scripts to stop functioning. At this stage the verification of script execution on changed pages is done manually, but we intend to automate script verification at least partially, in order to facilitate the task of volunteer helpers.

5 EMPIRICAL EVALUATION

We ran an empirical evaluation study with our first functional WNH prototype. The purpose of the study was three-fold: first, to find out if WNH can be of any help to functionally illiterate and blind users; second, to find out how WNH should be improved to suit their immediately perceived needs more appropriately; and finally, how can WNH be used to attend to yet a broader scope of interactive needs faced by these two similar but distinct groups of users.

5.1 Method and Procedures

As is often the case with new technologies, we chose a qualitative approach. Instead of trying to verify a set of hypotheses using statistical methods, we used interpretive methods [6]. By observing blind and functionally-illiterate adult users interact with WNH-see and WNH-read, respectively, and then interviewing them afterwards, we got hold of empirical evidence that broadened our understanding of the promises and challenges of WNH.

Because the perception of targeted users is likely to determine the tasks that volunteer helpers should do with WNH-support (and hence the very functionally and final architecture of the system), at this stage of our research we evaluated only WNH-see and WNH-read. We first ran two pilot tests with blind users and two pilot tests with functionally-illiterate users in order to fine-tune methodological procedures and the technical environment. Then we invited a group of six participants (three blind and three functionally illiterate) to run two different test scenarios followed by a short interview with open-ended questions. There were no participants challenged by both accessibility obstacles. Both groups were asked to visit government websites and try to solve a familiar problem situation for Brazilian citizens of their age.

In Table 1 we show a summary of the participants' profiles. We adopted a high variation sampling strategy for this study. For example, among blind participants there was considerable variation in age and familiarity with computers. Likewise, among functionally-illiterate participants, there was considerable variation regarding familiarity with computers, although less variation with respect to reading and writing abilities. Note that all participants had at least 7 years of schooling, which should mean (as was the case) that they were one step ahead of the rudimentary literacy level.

Before we discuss results we should quickly mention what happened during pilot tests (two users in each group). Blind pilot testers used WNH-see with the JAWS screen reader. Since there was no clear-cut boundary between navigation links on the sidebar (WNH-see interface) and the web page they should interact with through the agent they often got confused. Because of time-constraints for developing a new version of WNH-see that did not have this problem, we decided to handle it in a *Wizard of Oz* style: the experimenter *told* participants when they crossed the boundary. Functionally-illiterate pilot testers had difficulties understanding the initial sidebar interface. They got very confused with the script (which was visible on the left). Also, because

instructions provided orally about how to use WNH-read were insufficient, we decided to use a demo movie for the real test. These issues already point to methodological challenges imposed by functional illiteracy – breakdowns observed during the test may have stemmed both from technical *and* literacy problems interfering in the observers’ communication with the participants.

Table 1. Participants’ profiles in empirical study.

Participant	Age and History	Computer Literacy
P1 (WNH-see)	28 years old, blind for 27 years, 10% sight on the left eye	Uses computers about twice a week, with screen reader, since 2008
P2 (WNH-see)	24 years old, completely blind for 17 years	Uses computers about everyday, with screen reader, since 2003
P3 (WNH-see)	70 years old, born blind	Uses computers every two or three days, with screen readers, since 1994
P4 (WNH-read)	32 years of age, 10 years at school, reading and writing abilities near the upper threshold of basic liter- acy	Users computers almost every day, since 2005 (mostly to visit Youtube and Orkut)
P5 (WNH-read)	63 years of age, 12 years at school, reading and writing abilities near the upper threshold of basic liter- acy	Doesn’t care much about computers and the Inter- net.
P6 (WNH-read)	31 years of age, 7 years at school, basic reading and writing abilities	Frequent user of com- puters (twice a day) , web- sites most visited are Orkut (personal) and air- line sites (part of his work)

5.2 Results

None of the blind participants fully achieved the task proposed in the test scenario. Although all of them came very close to doing it, they had problems with following script execution. This was mainly because the narration text in WNH-see dialogs was poor. It just provided a ‘metric of progress’ (*e. g.* step 3 of 14), instead of something more informative like ‘you have successfully entered you Federal Revenue ID number’. Also, because of the mediation process, the session with the site often expired before the user could complete the dialog. Government sites used in the test had very tight expiration limits, probably for security reasons. Moreover, the mediating dialog voiced through JAWS during the test caused some confusion with the audio CAPTCHA. The transition between WNH-see and the visited website’s audio interface was problematic. Finally, because this is was a problem with the CoScripser version we used in the test, silent execution failures led the mediator interface to report success when it was *visible* that an error had occurred.

Results from tests with functionally-illiterate participants were considerably similar. Only one participant fully completed the proposed task (that, although cognitively

equivalent, was not the same as that for blind users). They also experimented session expiration problems, and were confused with CoScripter's sidebar and its silent failures. However, there were relevant differences. First, functionally-illiterate participants had considerable difficulty to understand explanations and instructions given both by us (experimenters) and WNH-read. Second, because they could see web page controls behind the mediator dialog (see Figure 3), they tended to jump to the page when they thought they had enough information. Although this also points to a transition issue between interacting with the mediator and with the web page (as for blind users with the audio CAPTCHA), it takes a completely different guise. Seeing participants seem to *want* mediation only to navigate more fastly through the pages. For example, these users don't need explanations to fill out their names, age, or address fields on typical forms. So, a mediator in this case is actually adding barriers rather than removing them.

Some of the evidence collected during the interviews show the value of our qualitative approach. A remarkable example comes from when one of the functionally-illiterate participants read WNH-read's instruction to type in the *letters* shown in the CAPTCHA image. He exclaimed: "But I see no letters!". This helped us understand a *literacy* issue with CAPTCHA. Recognizing transformed renditions of letter shapes may be a much harder test for functionally-illiterate users, even if they have no problem *seeing* the image. Moreover, because their listening skills may not be as developed as those of blind users, they may have problems with the audio CAPTCHA too. Therefore, new test alternatives should be explored for users in this condition.

6 CONCLUSIONS AND FUTURE WORK

We conclude that there is still a lot of work ahead of us with WNH-see and WNH-read, and that *it is worth doing*. As can be inferred from the above, collaborative script-based mediation for web navigation opens a large avenue for research in accessibility. Once the accessibility agent intercepts a user's goal-directed navigation on web pages, the communication between user and website can be organized and expressed in completely different ways. Compared to automatic agent mediation based on machine-learning and other artificial intelligent techniques, collaborative scripting has the advantage of putting humans in the loop. It taps on a number of skills and talents that some people have, like empathy, sensing and avoiding unproductive psychological attitudes from others, finding inspired analogies to explain more complicated things, handle cultural barriers, etc. These abilities, when found in volunteers, can actually lead to very fine-tuned adaptations of web interaction, not only for the communities targeted in this research, but for others as well (*e. g.* elderly users, people with motor disorders, etc.). This is, however, dependent on how WNH-support helps volunteers to design scripts and mediating dialogs for other users.

Our plans for future work include investigating interaction models that are specifically suitable for WNH-see or WNH-read and developing interactive templates to help volunteer scripters produce better scripts. Along these lines, we also intend to develop an interactive *mark-up language* to be used in scripts for specifying richer mediator dialogs. In Figure we show a sketch of what this mark-up language should look like. Note that it specifies the kind of dialogue that WNH is going to have with the user to achieve a particular goal – in this case getting a clearance certificate from the national income revenue service online.

```

3 <interaction>
4 System: "Este script vai ajudá-lo a tirar um comprovante de seu CPF.
5 Quando estiver pronto para iniciar, tecla ENTER.
6 Se quiser abandonar o script, tecla ESC.
7 User: {ENTER,ESC, explain('Explicações')}
8 Explain:
9 <interaction>
10 <!-- Escrever os diálogos de explicação aqui. -->
11 </interaction>
12 </interaction>
13 * go to "www.receita.fazenda.gov.br/"
14 * click the "Cadastro - CPF" link
15 * click the "Comprovante de Inscrição e de Situação Cadastral no CPF" link
16 <interaction>
17 System: "Estamos na página em que o sistema vai preencher por você o número
18 do seu CPF (que você já confirmou no bloco de anotações da janela ao
19 lado). Em seguida, você vai ter de digitar no teclado as palavras que
20 OUVIR na gravação que será tocada para você. Se você quiser repetir a
21 gravação, haverá uma opção para isto. Quando estiver pronto para
22 seguir adiante agora, tecla ENTER. Se quiser abandonar o script,
23 tecla ESC.
24 User: {ENTER,ESC, explain('Explicações')}
25 Explain:
26 <interaction>
27 <!-- Escrever os diálogos de explicação aqui. -->
28 </interaction>
29 </interaction>
30 * click the "Emissão do Comprovante" link
31 * enter your "CPF" into the "CPF:" textbox
32 <interaction> <!-- Escrever aqui o diálogo para o CAPTCHA --></interaction>

```

Figure 4. Sketch of a mark-up language for WNH assistive dialogs with users.

Finally, we also want to improve *housekeeping* tools like web bots that can automatically perform larger portions of script analysis and updates when the corresponding web pages change. These steps are necessary for us to reach more solid conclusions about the promises and challenges of collaborative web scripting for improved accessibility. Although the results of our empirical studies have pointed to more questions than answers, as is typical of exploratory research with innovative technologies, they have also shown the value and opportunity in this kind of research.

ACKNOWLEDGMENTS

We thank Allen Cypher for his many contributions to this research and test participants for their invaluable cooperation. Clarisse de Souza thanks CNPq for financial support and Chantal Intrator thanks CAPES for financing part of her MSc program.

REFERENCES

- [1] Aluísio, S. M., Specia, L., Pardo, T. A., Maziero, E. G., and Fortes, R. P. 2008. Towards Brazilian Portuguese automatic text simplification systems. In *Proceeding of the Eighth ACM Symposium on Document Engineering (Sao Paulo, Brazil, September 16 - 19, 2008)*. DocEng '08. ACM, New York, NY, 240-248.
- [2] Baloian, N., Luther, W., and Sánchez, J. 2002. Modeling educational software for people with disabilities: theory and practice. In *Proceedings of the Fifth international ACM Conference on Assistive Technologies (Edinburgh, Scotland, July 8-10, 2002)*. Assets '02. ACM, New York, NY, 111-118.
- [3] Bigham, J. P. and Ladner, R. E. 2007. Accessmonkey: a collaborative scripting framework for web users and developers. In *Proceedings of the 2007 international Cross-Disciplinary Conference on Web Accessibility (W4a) (Banff, Canada, May 07 - 08, 2007)*. W4A '07, vol. 225. ACM, New York, NY, 25-34

- [4] Bigham, J. P., Lau, T., and Nichols, J. 2009. Trailblazer: enabling blind users to blaze trails through the web. In Proceedings of the 13th international Conference on intelligent User interfaces (Sanibel Island, Florida, USA, February 8-11, 2009). IUI '09. ACM, New York, NY, 177-186.
- [5] Borodin, Y., Bigham, J. P., Stent, A., and Ramakrishnan, I. V. 2008. Towards one world web with HearSay3. In Proceedings of the 2008 international Cross-Disciplinary Conference on Web Accessibility (W4a) (Beijing, China, April 21 - 22, 2008). W4A '08, vol. 317. ACM, New York, NY, 130-131
- [6] Denzin, N.K. and Lincoln, Y.S. 1998. The landscape of qualitative research: Theories and issues. Sage Publications, Thousand Oaks, CA.
- [7] Devlin, S. and Unthank, G. 2006. Helping aphasic people process online information. In Proceedings of the 8th international ACM SIGACCESS Conference on Computers and Accessibility (Portland, Oregon, USA, October 23 - 25, 2006). Assets '06. ACM, New York, NY, 225-226.
- [8] Evett, L. and Brown, D. 2005. Text formats and web design for visually impaired and dyslexic readers—Clear Text for All. *Interacting with Computers* 17 (2005) 453–472
- [9] Harper, S. and Patel, N. 2005. Gist summaries for visually impaired surfers. In Proceedings of the 7th international ACM SIGACCESS Conference on Computers and Accessibility (Baltimore, MD, USA, October 09 - 12, 2005). Assets '05. ACM, New York, NY, 90-97
- [10] Intrator, C. 2009. Using Scripts to Improve Web Accessibility. Dissertação de Mestrado. Advisor: Clarisse Sieckenius de Souza - 2009. Departamento de Informática, PUC-Rio. 105 pages
- [11] Intrator, C. and de Souza, C. S. 2008. Using web scripts to improve accessibility. In Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems (Porto Alegre, RS, Brazil, October 21 - 24, 2008). ACM International Conference Proceeding Series, vol. 378. Sociedade Brasileira de Computação, Porto Alegre, Brazil, 292-295.
- [12] Leshed, G.; Haber, E.M.; Matthews, T.; Lau, T.A. 2008. CoScripter: Automating & Sharing How-To Knowledge in the Enterprise. In CHI Letters: Human Factors in Computing Systems, CHI 2008, 2008. 10(1): pp. 1719-1728
- [13] Little, G.; Lau, T.A.; Cypher, A.; Lin, J.; Haber, E.M.; Kandogan, R. 2007. Koala: Capture, Share, Automate, Personalize Business Processes on the Web. In CHI Letters: Human Factors in Computing Systems, CHI 2007, 2007. 9(1): pp. 943-946
- [14] Pontelli, E. and Son, T.C. 2003. Designing intelligent agents to support universal accessibility of E-commerce services. *Electronic Commerce Research and Applications* 2 (2003) 147–161
- [15] Ramakrishnan, I.V.; Mahmud, J.; Borodin, Y.; Islam, M.A.; Ahmed, F. 2009. Bridging the Web Accessibility Divide. *Electronic Notes in Theoretical Computer Science* 235 (2009) 107–124
- [16] Vanderheiden, G. 2008. Ubiquitous Accessibility, Common Technology Core, and Micro Assistive Technology: Commentary on “Computers and People with Disabilities”. *ACM Trans. Access. Comput.* 1, 2 (Oct. 2008), 1-7

- [17] WAI User Agent Accessibility Guidelines (UAAG) 2.0
[<http://www.w3.org/TR/2009/WD-UAAG20-20090723/>]
- [18] WAI Web Content Accessibility Guidelines (WCAG) 2.0
[<http://www.w3.org/WAI/intro/wcag>]
- [19] Walton, M. and Vukovic, V. 2003. Cultures, literacy, and the web: dimensions of information "scent". *interactions* 10, 2 (Mar. 2003), 64-71
- [20] Zajicek, M. and Edwards, A. (2004) Universal usability revisited (Editorial of IwC Special Issue on Computers and Universal Usability). *Interacting with Computers* 16 (2004) 403–410